THE ENHANCEMENT OF ARABIC INFORMATION RETRIEVAL USING ARABIC TEXT SUMMARIZATION

AHMAD HUSSEIN SALIM ABABNEH

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

August 2019

Table of Contents

List of Figures and Tables

Chapter 1 INTRODUCTION

Figures:

Tables:

Chapter 2 LITERATURE REVIEW

Figures:

Tables:

Chapter 3 METHODOLOGY

Figures:

Chapter 4 EXPERIMENTS and RESULTS

Figures:

Tables:

Chapter 5 EVALUATION and DISCUSSION

Figures:

Tables:

Statement of Publications Arising from This Thesis

The following publications have arisen from my research detailed in this thesis:

1. “Arabic Information Retrieval: A Relevancy Assessment Survey”, Ababneh, Ahmad, Joan Lu, and Qiang Xu, In: ISD2016 Proceedings. ISD, pp. 345-357. ISBN 9788378753070.

   Contribution of the candidate:

Conference Paper	ISD 2016 Poland
Paper Idea	Ahmad Ababneh
Data Collection	Ahmad Ababneh
Data Analysis	Ahmad Ababneh
Problem Design and Implementation	Ahmad Ababneh
Writing and Presentation	Ahmad Ababneb
Conference Attendance	Ahmad Ababneh
Revision and Supervision	Prof Joan Lu, Dr. Qiang Xu
The ratio of contribution for each author	
Ahmad Ababneh (First Author)	80%
Prof Joan Lu	15%
Dr. Qiang Xu	5%

2. “An efficient framework of utilizing the latent semantic analysis in text extraction”, Ababneh, Ahmad, Joan Lu, and Qiang Xu, International Journal of Speech Technology 22, no. 3 (2019): 785-815.

   Contribution of the candidate:

Journal Paper	International Journal of Speech Technology
Paper Idea	Ahmad Ababneh
Data Collection and Analysis	Ahmad Ababneh
Problem Design and Model Developement	Ahmad Ababneh
Programming and Implementation	Ahmad Ababneh
Results Collection and Evaluation	Ahmad Ababneh
Writing and Presentation	Ahmad Ababneb
Revision and Supervision	Prof Joan Lu, Dr. Qiang Xu
The ratio of contribution for each author	
Ahmad Ababneh (First Author)	80%
Prof Joan Lu	15%
Dr. Qiang Xu	5%

3. “An investigation into a new challenge in an efficient weighting scheme in VSM- based synonyms extraction”, Ababneh, Ahmad, Joan Lu, and Qiang Xu, Submitted to the Computational Intelligence Journal, 17-Sep-2019.

Contribution of the candidate:

Journal Paper	Computational Intelligence
Paper Idea	Ahmad Ababneh
Data Collection and Analysis	Ahmad Ababneh
Problem Design	Ahmad Ababneh
Programming and Implementation	Ahmad Ababneh
Result Collection and Evaluation	Ahmad Ababneh
Writing and Presentation	Ahmad Ababneb
Revision and Supervision	Prof Joan Lu, Dr. Qiang Xu
The ratio of contribution for each author	
Ahmad Ababneh (First Author)	80%
Prof Joan Lu	15%
Dr. Qiang Xu	5%

Abstract

The massive upload of text on the internet makes the text overhead one of the important challenges faces the Information Retrieval (IR) system. The purpose of this research is to maintain reasonable relevancy and increase the efficiency of the information retrieval system by creating a short and informative inverted index and by supporting the user query with a set of semantically related terms extracted automatically. To achieve this purpose, two new models for text mining are developed and implemented, the first one called Multi-Layer Similarity (MLS) model that uses the Latent Semantic Analysis (LSA) in the efficient framework. And the second is called the Noun Based Distinctive Verbs (NBDV) model that investigates the semantic meanings of the nouns by identifying the set of distinctive verbs that describe them.

The Arabic Language has been chosen as the language of the case study, because one of the primary objectives of this research is to measure the effect of the MLS model and NBDV model on the relevancy of the Arabic IR (AIR) systems that use the Vector Space model, and to measure the accuracy of applying the MLS model on the recall and precision of the Arabic language text extraction systems.

The initiating of this research requires holding a deep reading about what has been achieved in the field of Arabic information retrieval. In this regard, a quantitative relevancy survey to measure the enhancements achieved has been established. The survey reviewed the impact of statistical and morphological analysis of Arabic text on improving the AIR relevancy. The survey measured the contributions of Stemming, Indexing, Query Expansion, Automatic Text Summarization, Text Translation, Part of Speech Tagging, and Named Entity Recognition in enhancing the relevancy of AIR. Our survey emphasized the quantitative relevancy measurements provided in the surveyed publications. The survey showed that the researchers achieved significant achievements, especially in building accurate stemmers, with precision rates that convergent to 97%, and in measuring the impact of different indexing strategies. Query expansion and Text Translation showed a positive relevancy effect. However, other tasks such as Named Entity Recognition and Automatic Text Summarization still need more research to realize their impact on Arabic IR.

The use of LSA in text mining demands large space and time requirements. In the first part of this research, a new text extraction model has been proposed, designed, implemented, and evaluated. The new method sets a framework on how to efficiently employ the statistical semantic analysis in the automatic text extraction. The method hires the centrality feature that estimates the similarity of the sentence with respect to every sentence found in the text. The new model omits the segments of text that have significant verbatim, statistical, and semantic resemblance with previously processed texts. The identification of text resemblance is based on a new multi-layer process that estimates the text-similarity at three statistical layers. It employes the Jaccard coefficient similarity and the Vector Space Model (VSM) in the first and second layers respectively and uses the Latent Semantic Analysis in the third layer. Due to high time complexity, the Multi-Layer model restricts the use of the LSA layer for the text segments that the Jaccard and VSM layers failed to estimate their similarities. ROUGE tool is used in the evaluation, and because ROUGE does not consider the extract's size, it has been supplemented with a new evaluation strategy based on the ratio of sentences intersections between the automatic and the reference extracts and the condensation rate. The MLS model has been compared with the classical LSA that uses the traditional definition of the singular value decomposition and with the traditional Jaccard and VSM text extractions. The results of our comparison showed that the run of the LSA procedure in the MLS-based extraction reduced by 52%, and the original matrix dimensions dwindled by 65%. Also, the new method achieved remarkable accuracy results. We found that combining the centrality feature with the proposed multi-layer framework yields a significant solution regarding the efficiency and precision in the field of automatic text extraction.

The automatic synonym extractor built in this research is based on statistical approaches. The traditional statistical approach in synonyms extraction is time-consuming, especially in real applications such as query expansion and text mining. It is necessary to develop a new model to improve the efficiency and accuracy during the extraction. The research presents the NBDV model in synonym extraction that replaces the traditional tf.idf weighting scheme with a new weighting scheme called the Orbit Weighing Scheme (OWS). The OWS weights the verbs based on their singularity to a group of nouns. The method was manipulated over the Arabic language because it has more varieties in constructing the verbal sentences than the other languages. The results of the new method were compared with traditional models in automatic synonyms extraction, such as the Skip-Gram and Continuous Bag of Words. The NBDV method obtained significant accuracy results (47% R and 51% P in the dictionary-based evaluation, and 57.5% precision using human experts' assessment). It is found that on average, the synonyms extraction of a single noun requires the process of 186 verbs, and in 63% of the runs, the number of singular verbs was less than 200. It is concluded that the developed new method is efficient and processed the single run in linear time complexity (O*(n)*).

After implementing the text extractors and the synonyms extractor, the VSM model was used to build the IR system. The inverted index was constructed from two sources of data, the original documents taken from various datasets of the Arabic language (and one from the English language for comparison purposes), and from the automatic summaries of the same documents that were generated from the automatic extractors developed in this research.

A series of experiments were held to test the effectiveness of the extraction methods developed in this research on the relevancy of the IR system. The experiments examined three groups of queries, 60 Arabic queries with manual relevancy assessment, 100 Arabic queries with automatic relevancy assessment, and 60 English queries with automatic relevancy assessment. Also, the experiments were performed with and without synonyms expansions using the synonyms generated by the synonyms extractor developed in the research.

The positive influence of the MLS text extraction was clear in the efficiency of the IR system without noticeable loss in the relevancy results. The intrinsic evaluation in our research showed that the bag of words models failed to reduce the text size, and this appears clearly in the large values of the condensation Rate (68%). Comparing with the previous publications that addressed the use of summaries as a source of the index, The relevancy assessment of our work was higher than their relevancy results. And, the relevancy results were obtained at 42% condensation rate, whereas, the relevancy results in the previous publication achieved at high values of condensation rate. Also, the MLS-based retrieval constructed an inverted index that is 58% smaller than the Main Corpus inverted index.

The influence of the NBDV synonyms expansion on the IR relevancy had a slightly positive impact (only 1% improvement in both recall and precision), but no negative impact has been recorded in all relevancy measures.

Keywords

Automatic Text Extraction, Automatic Synonyms Extraction, Cosine similarity, Information Retrieval, Multi-layer Similarity, Natural Langauge Processing, Latent Semantic Analysis, Orbit Weighting Scheme, Vector Space Model.

Abbreviations

AF	Average f-score
AIR	Arabic Information Retrieval
AP	Average Precision
AR	Average Recall
ASE	Automatic Synonyms Extraction
ATE	Automatic Text Extraction
ATS	Automatic Text Summarization
CBoW	Continuous Bag-of-Words
CLIR	Cross-Language Information Retrieval
CR	Condensation Rate
FULLC	Full Containment
GIR	Geographical Information Retrieval
HIGHC	High Containment
idf	Invert Term Frequency or Document Frequency
IR	Information Retrieval
LOWC	Low Containment
LSA	Latent Semantic Analysis
MAP	Main Average Precision
MLS	Multi-Layer Similarity
MODC	Moderate Containment
MT	Machine Translation
MRD	Machine-readable Dictionary
NBDV	Noun Based Distinctive Verbs
NER	Named Entity Recognition
NLP	Natural Langauge Processing
NN	Neural Network
OWS	Orbit Weighting Scheme
P	Precision
POS	Part of Speech Tagging
PRF	Pseudo-Relevance Feedback
QE	Query Expansion
R	Recall
RF	Relevance Feedback
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RSI	Ratio of Sentences Intersections
SEE	Summary Evaluation Environment
SG	Skip-Gram model
SVD	Singular Value Decomposition
tf	Term Frequency
tf.idf	the weighting scheme based on tf and idf
VSM	Vector Space Model

CHAPTER 1 INTRODUCTION

Information Retrieval (IR) is the field of computer science that establishes a framework on how to store, represent, and retrieve text documents from a huge collection of unstructured text. The IR process includes a series of operations that include text processing, index creation, statistical weighting and matching, text retrieval, and documents ranking. IR plays an essential role in all aspects of life, and any improvements achieved on the IR makes the information extracted from the internet more relevant.

Automatic Text Summarization (ATS) is the computer's ability to simulate human beings' skills in drawing the saliant ideas or the key points from a particular text. The summary represents important information found in the text. If the summary of a text document contains sufficient information, it can be employed in place of the full document itself in the IR systems. In this research, we integrated the IR with a semantic ATS model. We satisfy the equation that keeps the relevancy of the IR system reasonable, and at the same time, we aim to reduce the retrieval time. Thus, we aim to improve the AIR system performance by building a small and informative index. The index size is reduced using the ATS method that produces a variable-sized summary and keeps all the main terms reserved.

The introduction chapter includes eight essential sections, section 1.1 gives a brief introduction of the Information Retrieval field and depicts the place of our semantic ATS models in the structure of the IR process, section 1.2 explains the IR operations addressed in this research, and section 1.3 defines the relevancy measures that are normally used in evaluating the IR system output. Section 1.4 introduces the ATS concepts and the NBDV model, and section 1.5 discusses the reasons for using the Arabic Language as a case study language. Section 1.6 lists the research aim and objectives, and section 1.7 summarizes the main contributions of the research.

1.1 Introduction

IR aims to retrieve relevant documents that match the user information need (Baeza-Yates & Ribeiro, 2011). As showing in Figure 1.1, the IR system accepts two kinds of data, the user query that reflects the user information need, and a collection of a huge number of documents stored in the computer memory. Depending on the user

information need, the retrieved documents are normally classified as relevant or irrelevant documents. The relevant documents are a subset of the documents found on the internet or in a specific corpus that satisfy the user information need.

Figure 1.2 explains the IR process, and it abstracts the process in four steps. In step one, the system pre-processes the documents and the user query and makes necessary changes such as: filtering the text from the punctuation marks or special characters, tokenizing the text to sets of terms, and deleting the Stopwords term (Ababneh, Kanaan, Al-Shalabi, & Al-Nobani, 2012). In step two, the filtered terms are transferred to the indexing step in which the system builds a unified representation for all the documents. The purpose of this representation is to specify the importance of each term with respect to each document. The most popular example of such representation is the inverted index. The inverted index links each term to the documents that contain it, and the link appears as a score that represents the importance of this term to the linked document. In step three, a certain IR model is used to match the query terms with the index terms to find the relevant documents. Boolean model, Vector Space model, and the probabilistic models are the most widely used. The output of the matching process is a set of ranked retrieved documents. Finally in step four, the user can judge the output set and make the necessary feedback that may necessitate the change of the query terms, and the process initiated again until the user need is satisfied. This abstraction hides a lot of details, but we used it to pinpoint the main steps in the Information Retrieval process.

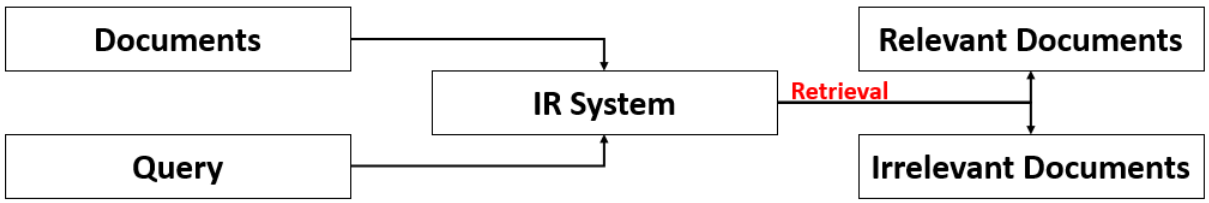


Figure 1.1 IR System – Input/Output

It's important to mention in this context that the IR system is not a database system that has an exact answer to every item stored in the database. The IR system deals with text documents that do not have a well-established structure. This implies that any information retrieval system returns a set of documents and the judgment of whether

the answer set provided by the IR system is relevant or not depends on the user who can decide that. Thus, only the user can determine the degree of relevancy that reflects his satisfaction.

The IR collects large text documents and stores them in the inverted index. On the user search, the inverted index is inspected to make the necessary matching between the user terms and the index terms. One of the major problems facing the researchers in the IR field is the massive growth of the text volume on the internet. In 2018, the worldwide web-size site (Maurice, 2018) published that the number of indexed pages on the internet reached 5.28 billion pages. This huge volume of text imposes to find an innovative solution to store and retrieve the text data efficiently.

Automatic Text Summarization is a computerized process of condensation that yields a shorter version of the original text and keeps the core meaning and the main ideas reserved (Meena & Gopalani, Domain Independent Framework for Automatic Text Summarization, 2015). To solve the problem of text data overload, we enhanced the solutions proposed in (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001), and (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013). The authors of those solutions tried to reduce the inverted index by using the document summaries that are generated automatically by the ATS systems. The summaries are used as the source of the index instead of the original documents. Two reasons impede the use of ATS as a supporting tool to enhance information retrieval performance. The first one related to the methods used to extract the summaries, in (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001), and (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013), the authors used statistical techniques based on traditional parameters such as the term frequency and the term distribution that are originally proposed in the field of IR to weight the documents and query terms in the VSM model. These techniques did not consider the semantic meaning of the text and produced low quality summarizes that hurt the recall of the developed IR system (Sakai & Sparck-Jones, 2001). The second reason that impedes the use of the ATS in the IR system is the time banality of applying advanced statistical techniques that use the semantic analysis to summarize the documents such as the LSA model. The LSA is a statistical model that can simulate the way people acquire knowledge and meaning through the correlation of facts from several sources (Ngoc & Tran, 2018). The LSA is proposed in the literature to solve the VSM semantic problems (Yates & Neto, 1999), but the time consumption of the LSA is the challenge (He, Deng, & Xu, 2006). Therefore, an improved solution

of generating the summaries should be designed to be: (1) feasible for the IR indexing system (from the time and contents perspectives), (2) effective from the IR relevancy perspective (retrieve the desired relevant document).

To solve the two problems discussed in the previous paragraph, we proposed to equip the IR system with two models (Figure 1.2). The first one is the Multi-layer Similarity (MLS model), which is a text extractor that uses a multilayer approach of statistical analysis with a semantic investigation in complicated cases to generate a condensed version of the inverted index. The MLS extractor uses the LSA in the case that the verbatim similarity and the VSM similarity obtain low similarity results. The structure of the MLS extractor described in detail in chapter 3. The second model is the Noun Based Distinctive Verbs synonyms extractor (or the NBDV extractor). The NBDV model is used to extract the synonyms of the query terms. The purpose of the NBDV extract is to fix any loss of information caused by the MLS extractor and to expand the semantic investigation of the user query terms.

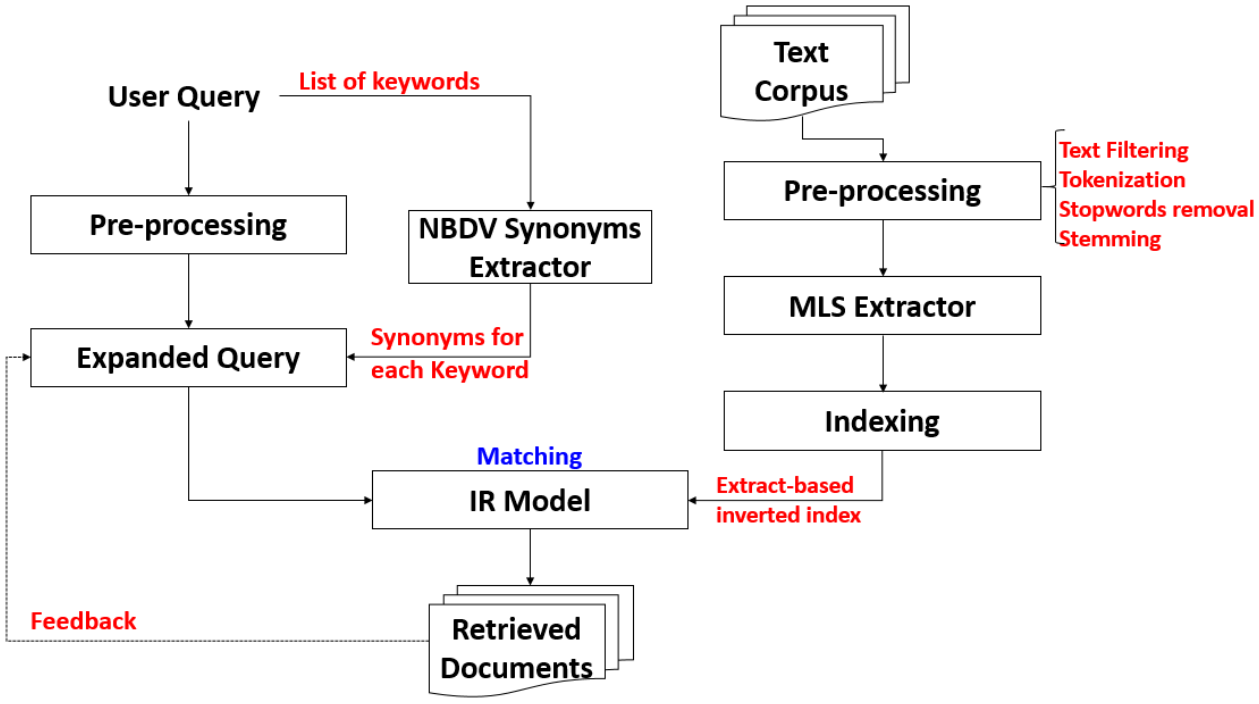


Figure 1.2 Proposed IR System

1.2 The IR Operations Investigated in the Thesis.

In this research, two essential IR operations were investigated, the indexing and the query expansion.

1.2.1 Indexing

In Information Retrieval, the index is a data structure (in the form of a linked list or hash table) that represents the contents of the document as a set of weighted stems or roots (Tokens) (Baeza-Yates & Ribeiro, 2011). The index is built after important preprocessing operations; Tokenization, Stopwords removal, and stemming. The mentioned preprocessing operations add value in terms of effectiveness and efficiency for the IR system.

The IR system does not scan the contents of each document sequentially. The index terms are matched against the user query terms to find the terms inside the index that best match the user query. Therefore, the well-established index is necessary to optimize the retrieval speed and performance (Abderrahim, Mohammed, & Mohammed, 2016).

The entries of the index can be single words (SW) in forms of stems, roots, or full words, or multi-words (MW) in forms of short phrases that improve the semantic side of the index (Boulaknadel, Daille, & Driss, 2008). Semantic indexing requires word senses disambiguation and tries to link the index term to the meaning that suits the context. Researchers have examined different strategies to identify suitable indexing strategies to improve the precision measure.

The inverted index is the most popular representation of the documents and queries in modern IR applications. The organization of the information inside the inverted index supports the fast search (Baeza-Yates & Ribeiro, 2011). The inverted index contains two main parts, the vocabularies (terms) and the posting list. The vocabularies are stored in a lexicographical order. Each term has one posting list, and each entry in the posting list contains the identification number of the document that contains the term with the number of times the term appeared in the document. In the VSM model, the posting list information is transferred to quantitative scores that represent the weights of the term in each document (Baeza-Yates & Ribeiro, 2011).

The retrieval speed depends on the type of data structure used to implement the index, for example, if the hash structure is used, then the searching requires O(1), and if the tree structure is used, then the search requires O(c), where c is the length of the term (Baeza-Yates & Ribeiro, 2011).

The problem related to the use of the inverted index is the space it occupied in the disk, and the insertion, deletion, and update operation (Patil, et al., 2011), (Baeza-Yates & Ribeiro, 2011). Lin and Chris in (Lin & Chris, 2010) stated that the space overhead of the inverted index varies and unpredictable (it depends on the contents of the posting list). To solve the problem of the space overhead, we proposed to reduce the size of the original documents before the indexing step is initiated, and this reduction can be performed by extracting the salient components of the documents by efficient and accurate text extraction system and using these components as a source of the index.

1.2.2 Query Expansion and Relevance Feedback

Query Expansion (QE) is an IR technique used to improve the relevancy of IR systems through the amendment of the query terms. Normally, the user query is short and contains a few numbers of terms, and sometimes the ignorance of the subject being searched makes the searcher uses inadequate query terms. The QE assumes that the searcher query is the problem in retrieving irrelevant documents, and the relevancy can be improved by either automatically adding new query terms to the query or by substituting the query terms with new ones.

Two main strategies have been employed to expand the searcher's query; the first one uses linguistic approaches to add synonyms or semantically related words to the terms that are mentioned in the query and drives. The second method uses automatic user feedback to improve the query terms. Another categorization of QE strategies is Global vs. Local. Global methods expand the query by hiring external resources such as a dictionary of synonyms, lexicon thesaurus, or stemming algorithm. Local methods collect important terms from the top-ranked documents that appear in the first run of the information retrieval system and use them to expand the query terms (Christopher, Prabhakar, & Hinrich, 2009).

Relevance Feedback (RF): is a query expansion strategy that involves the enhancement of the query terms by adding new terms taken from the top-ranked retrieved set achieved in the first search. After the first search, the user or the system specifies the set of documents that seem to be relevant; then, a relevance feedback algorithm is used (such as the Rocchio algorithm) to make the necessary enhancement. The RF is very useful in the case that the user does not have enough information about the problem domain. Pseudo RF, or (PRF) for short, is the automated version of the RF in which the IR system automatically retrieves a set of terms from the top-ranked document and refines the user query.

All the publications reviewed in this research showed a positive impact of QE in the relevancy of the information retrieval systems. The recall and precision were improved; the ratio of improvement in average precision lies between 1% in (Hanandeh, 2013) and 14% in (Wedyan, Alhadidi, & Alrabea, 2012). The precision showed either enhanced (Abdelali, Cowie, & Hamdy, 2007), (Atwan, Mohd, Rashaideh, & Kanaan, 2016), (Mallat, Anis, Emna, & Mounir, 2013), (Wedyan, Alhadidi, & Alrabea, 2012) or stable ratio (Abbache, Barigou, Belkredim, & Belalem, 2016), (Hanandeh, 2013). This improvement provides a strong indication that the query expansion obtained more relevant documents and omitted some of the irrelevant ones. However, the traditional statistical methods to automatically generate the expansion terms are time-consuming (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012), (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), and (Benabdallah, Abderrahim, & Abderrahim, 2017). One of the Objectives of this research is to obtain informative expansion terms by developing an efficient and accurate statistical synonyms extraction model.

1.3 IR Relevancy Evaluation

The evaluation of the information retrieval system is a very complicated process. Measuring the Effectiveness of the IR application means evaluating the relevancy, which is seen from the user's point of view. The performance of such applications in terms of time and memory is not easy to predict because of the growing volume of data over the internet. Gold corpus or Dataset - can be helpful in this regard and can be used to make the required comparison between different proposed systems. The corpus contains a fixed number of documents, a set of queries, and a manual matching between the queries and documents to determine the relevance set of the documents for each query. However, this kind of judgment is binary and does not reflect the real situation, and a huge effort is required to build such a corpus (Christopher, Prabhakar, & Hinrich, 2009).

Table 1.1 summarizes the most important relevancy measurements used in IR [(Baeza & Ribeiro, 1999), (Christopher, Prabhakar, & Hinrich, 2009)]. The unranked retrieval measurements give a general indication about the contents of the retrieved set, whereas, the ranked retrieval measurements gives a strong indication about the quality of the answer set and magnifies the retrievals that rank the relevant documents on the top of the answer list.

1.4 Text Mining Strategies Investigated in the Thesis

The introduction section of this chapter proposed our idea to enhance IR relevancy and efficiency. Mainly, the proposed enhancement depends on increasing the efficiency and accuracy of two text mining principles; the

Automatic Text Summarization and the Automatic Synonyms Extraction (ASE). The ASE is an extraction system that automatically investigates the word synonyms. The ATS is used to create condense and informative inverted index and the ASE is used to equip the user query with semantically related terms. New and efficient ATS and ASE models are developed, and we measured their effect on a real IR system developed for the Arabic language.

Table 1.1 IR Evaluation Measures

Precision (P)	The number of retrieved documents that are relevant, divided by the number of retrieved documents.	Unranked Retrieval Evaluation
Recall (R)	The number of retrieved documents that are relevant, divided by the total number of relevant documents.	Unranked Retrieval Evaluation
Average Recall (AR)	The Average of all recall values that are obtained for n queries	Unranked Retrieval Evaluation
f-score (F)	The harmonic mean of R and P $F = \frac{2\ P\ R}{P\ +\ R}$	Unranked Retrieval Evaluation
Average F-Score (AF)	The Average of all f-score values that are obtained for n queries	Unranked Retrieval Evaluation
R-th Precision	The precision value at a specific position in the retrieved ranked list. (the number of the relevant document should be known in advance).	Ranked Retrieval Evaluation
Average Precision (AP)	Computed for each query, in which we compute the precision when each relevant document is retrieved, then we compute the average of all precision values obtained.	Ranked Retrieval Evaluation
Mean Average Precision (MAP)	Computed for all queries, equals the average of AP	Ranked Retrieval Evaluation
Interpolated Average Precision	Traces the maximum precision at 11 recall levels, Ri = {0.0, 0.1, 0.2 , 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, R0=0.0, R1=0.1... and R10=1.0 $P(R_i) = \max_{i \le r \le i+1} P(R)$ This measure answers the question; what is the maximum precision value achieved when the recall values ranged between x and y?	Ranked Retrieval Evaluation

1.4.1 Automatic Text Summarization

Every day, the world uploads a huge amount of text in forms of articles, books, emails, tweets, blogs, and others. In 2010, The Compare Business Product website[1] published that the database of the World Data Centre for Climate stores 220 terabytes of text, and the Library of Congress contains 5,000,000 digital documents (20 terabytes of text), and every day, the library records 10,000 items. In 2017, the World Wide Web website[2] published that the number of pages indexed by Google and Bing search engines reached 4.61 billion pages. This massive volume of data requires a huge inverted index that consumes a large main memory and disk space, and it increases the processing time of the user's need. Therefore, two reasons necessitate the development of efficient automatic text summarizer, the huge growth of text data and the need to obtain accurate and fast results.

Automatic Text Summarization or ATS is a computer-based text condensation process that has been studied since the fifties of the previous century (Luhn, A Statistical Approach to Mechanized Encoding and Searching of Literary Information, 1957), (Luhn, The Automatic Creation of Literature Abstracts, 1958). The ATS aims to produce a short version of the original text and keeps the salient ideas reserved (Kupiec, Pedersen, & Chen, 1995), (Mani, Automatic Summarization., 2001), (Binwahlan, Salim, & Suanmali, 2009), (Mei & Chen, 2012), (Nenkova & McKeown, A Survey of Text Summarization Techniques, 2012), (Meena & Gopalani, Domain Independent Framework for Automatic Text Summarization, 2015).

In this work, we concentrate on two kinds of ATS, the Automatic Text Extraction (ATE) and the Automatic Synonyms Extraction. The ATE copies the salient parts of the text without adding any information or changing the text structure.

The ATE is used to reduce the index size in this research (see (Mani, Automatic Summarization., 2001)). The selection of the ATE as a reduction tool of the main inverted index was based on two reasons:

1. Simplicity: the ATE copies certain parts of the text without any further syntactic and lexical operations.
2. Adequacy: unlike the abstraction methods in ATS, The ATE reserves the terms and concepts mentioned in the original document. This merit is essential in this investigation because it allows us to compare the relevancy of the original text inverted index and the summaries inverted index.

[1] Http://www.comparebusinessproducts.com/fyi/10-largest-databases-in-the-world (Seen in 8-8-2017)
[2] http://www.worldwidewebsize.com (Seen in 8-8-2017)

As mentioned previously, The ATE model developed in this investigation is called MLS extraction, and it is built based on a multi-layer hierarchy in which several statistical approaches were employed in an incremental manner.

1.4.2 Automatic Synonyms Extraction

Besides the Multi-Layer similarity model for text extraction, the research proposed here presents the Noun Based Distinctive Verbs synonym extraction model that is based on a new weighting scheme called Orbit Weighting Scheme. The OWS is used in the weighting phase of the NBDV method to replace the traditional tf.idf weighting scheme used in the skip-gram model or the Continuous Bag-of-Words model (Mikolov, Chen, Corrado, & Dean, 2013). The OWS uses the verbs to weight the nouns. The OWS is designed for nouns because the nouns are the primary concern of the text mining applications, mostly, all the query terms in Information retrieval, the class and category names in text categorization, the concept/entity in entity recognition, and others are nouns.

As stated by Webb, the languages are rich in synonyms or semantically related words that give the writer the ability to describe the same entity using different words and yield interesting and more vivid text (Webb, 2007). However, the use of synonyms confuses the text mining systems that employ the exact matching approaches such as the Boolean models or the statistical models that use the term frequency and the term distribution to determine the importance of the terms (Schütze, Manning, & Raghavan, Introduction to information retrieval, 2008). Therefore, in the text mining fields, it is necessary to develop Automatic Synonyms Extraction systems to identify the synonyms for the text mining applications.

In many Natural Language Processing (NLP) and IR publications, the semantic investigation of the text contents was improved by hiring a semantic dictionary (such as the synonyms dictionary) in the investigation process (Barak, Dagan, & Shnarch, 2009), (Dinh & Tamine, 2015), (AlMaayah, Sawalha, & Abushariah, 2016). The semantic dictionaries with what they contain of synonyms are evaluated as valuable tools. These tools improve the precision and the recall of the NLP and IR applications, for example, in the field of text categorization, in (Barak, Dagan, & Shnarch, 2009), the recall increased from 71% to 92%, and in the field of Information retrieval, in (Dinh & Tamine, 2015), the MAP increased by 5.61%. Also, The ASE supports the term weight, which is necessary to determine the importance of the word in a particular context (AlMaayah, Sawalha, & Abushariah, 2016).

The precision and efficiency are the primary concerns of any statistical synonyms extraction systems, the precision measures the ratio of correctness in the answer set, and the efficiency measures the time and space penalty. Recently, important publications in the ASE field used statistical approach and gained significant precision (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012), (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), and (Benabdallah, Abderrahim, & Abderrahim, 2017), but they did not consider the efficiency; and the time required for their systems tends to be long. For example, Leeuwenberga et al. in (Leeuwenberga, Vela, Dehdar, & Genabith, 2016) used a bag of word model called relative cosine similarity to extract the term synonyms. In their work, the construction of the terms-terms weighted matrix is expensive in terms of space and time, and the repetitive computations add more delay. Also, Minkov and Cohen (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012) used a path constrained graph, and the problem with this graph is the high time required to construct the graph and the space needed to store the graph. The graph stores each term in the corpus with all existing edges that link this term to the other terms found in the corpus, add to this the time needed to follow all the paths that lead to the terms. Henriksson et al. (Henriksson, Moen, Skeppstedt, Daudaravicius, & Duneld, 2014) manipulated two efficient distributional hypothesis models over two large corpora to investigate more semantic relations between the terms, but even that the recall was reasonable (47%), it was obtained in a very low precision (8%). This means that the answer set size was very large, and only 8% of the answer was correct. Henriksson et al. used the random permutation and random indexing to construct the required semantic spaces, and those two techniques do not have the accuracy of more advanced statistical techniques such as the Latent Semantic Analysis. It maps the terms-documents space to a terms-concepts space or a concepts-documents space. The time complexity of the LSA is very high because the LSA captures the meaning of the term by creating a sophisticated network in different contexts and huge datasets (He, Deng, & Xu, 2006).

Replacing the traditional tf.idf weighting scheme with a new high-performance weighting scheme in the VSM-based text extraction is the main contribution of the NBDV model. The traditional statistical approach in synonyms extraction is time consuming especially in real applications such as the query expansions and text mining (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012), (He, Deng, & Xu, 2006), and (Mikolov, Chen, Corrado, & Dean, 2013). The tf.idf weighting scheme is used in the models that adapted the vector space model to be applicable to work in synonyms extraction, such as the skip-gram model (SG) or the Continuous Bag-of-Words (CBoW) models. In the field of synonyms extraction, The SG model is used to predict

the source context words of a given word, whereas the CBoW is used to identify the missing word given the context words. In SG and CBoW models, a vector of weights is created for each term in the text. This vector consists of the weights of every term in the corpus with respect to the term being processed, which yields a number of weighting processing steps equal to the square number of the terms. (Mikolov, Chen, Corrado, & Dean, 2013). Therefore, enhancing the weighting scheme is an important issue in synonyms extraction methods to improve the speed of producing the vectors of the terms.

A new method is needed to improve the efficiency and accuracy during the extraction. This research discusses the efficiency and accuracy enhancement of the synonyms extraction through the amendment of the traditional weighting schemes. One of the objectives of this investigation is to obtain fast and accurate synonyms extraction by using an enhanced weighting scheme in the weighting phase of the vector-space based synonyms extraction process. The enhanced weighing represents a developed version of the traditional tf.idf that remedies the time penalty of weighting the contents of the text.

1.5 Linguistic Issues and Challenges

The Arabic language is the language used in this research as a case study. The Arabic language is the first language of 450 million people in the Arab country, and the religious language the Muslims around the world (Arabs, 2019).

The Arabic language is one of the Semitic languages. The number of letters used to form its vocabularies is 28, and the language is written from right to left. The Arabic language has three variations; Classical, Modern, and Colloquial Arabic. Classical Arabic is the language of Al-Quran and literature. The Modern Arabic language (MAL) uses simple vocabularies and this feature participates in making the MAL easy to understand. In Arab countries, MAL is the language of official documents, educational institutes, and media. Dialectal or Colloquial Arabic is a form of the Arabic language that in use in everyday talk (Alshamrani, 2012).

The Arabic words are constructed from the root, and the Arabic Language is classified as inflectional and derivational language, which means that a large number of words can be constructed from the single root. For example, 48 variations can be obtained from the Arabic root "درس" means "learn". Besides the prefix and postfix, the Arabic language has infixes in the middle of some words. For example, "لاعب" player" has "ا" as infix and "حقول" fields" has "و" as an infix.

The processing of the Arabic language is a real challenge due to the existence of distinguishing features that complicated the process of the Arabic text. Ryding in (Ryding, 1991) mentioned these features: (1) No short vowels; and instead, the Arabic language writers use special characters called diacritics that are usually discarded by the writers because they assume that their meanings can be captured from the context. (2) No Capitalization, which increases the complexity of extracting the entities that are mentioned in a given text. (3) in some cases, Arabic allows the construction of the sentence without an implicit or explicit subject. Accordingly, the sentence ("the house was demolished" هُدم البيت) is a complete sentence even though it does not contain a clear subject.

One of the primary objectives of this research is to measure the relevancy effect of the Multi-Layer Similarly model on the IR systems that uses the Arabic Language as a case study and to measure the accuracy of applying the MLS model on the recall and precision of the Arabic language text extraction.

To achieve the objectives that are mentioned in the previous paragraph, we employed three well-known datasets for the Arabic Language in a series of experiments.

1. Essex[3] Corpus: The corpus contains 153 Arabic articles with UTF and ISO formats. Five manual extracts have been produced for each article. The corpus contains documents with different subject areas. From our references list, the corpus has been used recently by Al-Radaideh and Bataineh in (Al-Radaideh & Bataineh, 2018).
2. Kalimat[4] Corpus: This corpus contains 20,291 Arabic article (3,537,677 Noun, 1,845,505 Verb, 115225 adjectives, and totally 6,286,217 terms). The corpus comprises greater than 6,000,000 terms. The data was taken from Omani newspapers. The corpus contains documents with a variety of subjects. The Kalimat subjects include health, science, history and art, religion, technology, environment, economic, and financial aspects (Li, Forascu, El-Haj, & Giannakopoulos, 2013) (Al-Radaideh & Bataineh, 2018).
3. 242-documents: The corpus contains 60 queries with their manual relevancy assessment, the source is the Saudi National computer conference, the domain is computer science and informatics, used by (Ghwanmeh, Kannan, Riyad, & Ahmad, 2007) (Hanandeh, 2013).

1.6 Research Aim and Objectives

This research aims to improve the efficiency and performance of the information retrieval systems. We examine if the employment of Automatic Text Extraction can give information retrieval systems the ability to obtain more

[3] Can be downloaded from http://www.lancaster.ac.uk/staff/elhaj/corpora.htm

[4] Can be downloaded from: https://sourceforge.net/projects/kalimat/

relevant documents in a short time. Text summarization is one of the important NLP principles that can investigate the document's contents by extracting the salient parts that represent the core of the document.

The research achieves the following objectives:

1. Performing a precise survey that reviews the important publications in AIR and provides a starting point for new researches in this field.
2. Setting a framework on how to employ the statistical semantic analysis based on the efficient use of the Latent semantic analysis in the text extraction.
3. Building an effective text summarizer using the efficient framework of semantic analysis.
4. Proving that the use of the traditional statistical bag of word models (such as theVSM and Jaccard coefficient) is not suitable for performing reasonable text summarization, especially to reduce the inverted index in an IR system.
5. Improving the retrieval time through the reduction of the index size, which will be constructed from the summaries instead of the original documents.
6. Analyzing the relevancy measures of the Information Retrieval systems with and without Automatic Text Summarization using IR evaluation measures.
7. Developing an efficient synonyms extraction model and employ this model in a synonyms extraction system that extracts the synonyms of the user query terms.
8. Enhancing the user query with the synonyms generated automatically and testing their impact on the IR system that uses the summaries as a source of the index.
9. Estimating the effectiveness of our summarizers using extrinsic methods by evaluating their influence on the AIR system.
10. Comparing the results of employing the Arabic text summarization in information retrieval with previous results that have been obtained on other languages such as English.

1.7 Main Contributions

The main contributions of this research are summarized in the following points:

1. Efficient and informative inverted index using a semantic-based text summarizer. Firstly, No actual work was found for the employment of ATS in IR systems that had built to retrieve information for the Arabic language, and we have little work that investigated the impact of ATS on the IR performance for the other languages. All the found research publications improve the precision and hurt the recall. Brandow et al. in (Brandow, Karl, & Lisa, 1995) obtained a high precision rate when they used domain-independent automatic summaries (extractive summary), based on the traditional tf.idf statistical sentence extraction, as index source. They evaluated the results obtained from their automatic extracts and compared them with simple extracts whose sentences were copied from the first few sentences in the original document (called Lead summary) and also with full-text indexing, and they tested the relevancy against three condensation rate 60, 150, 250 words, and they obtained the highest precision at 150 and 250 Condensation Rate (CR equals the summary length divided by the full-text length), but the recall decreased from 100% to 59%. In our research we showed that the traditional weighting schema based on tf.idf with cosine similarity is not a significant tool for text extraction, and it obtains high relevancy assessment, but it fails to delete a reasonable portion of the text (this is true in the experiment of Ronald because he obtained the highest precision at the largest comparison rate 250 words). The same conclusion can be driven from Sakai and Sparck-Jones (Sakai & Sparck-Jones, 2001), they used the summary-based corpus as a source of indexing, but they evaluated the impact of generic extracts with or without PRF. We are considering the generic summaries as a source of indexing because we do not use the PRF because it is a query refinement strategy and out of the scope of this research. The results of Sakai and Sparck-Jones in (Sakai & Sparck-Jones, 2001) were in line with the results obtained in (Brandow, Karl, & Lisa, 1995), in which a summary-only as source of index obtained the largest precision at the highest condensation rate (see table 5 and 6 in (Sakai & Sparck-Jones, 2001)). Another important thing raised the precision in (Sakai & Sparck-Jones, 2001) is the inclusion of the title of the document in the index and the exclusion it from the CR computation, and the title mainly conveys important terms or concepts. In Geographical Information Retrieval (GIR), which represents an IR system for retrieving geographical information from unstructured text (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013), Perea-Ortega et al. in (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013) utilized two kinds of summaries; automatic general summary based on statistical parameters such as the tf and existence of noun phrases, and automatic geographical summary, which gave more attention to the sentences that refer to the geographical entities that were mentioned in the text. Perea-Ortega et al. gave a clear conclusion that the use of statistical single document summarizes as the source of indexing is not significant. Our work is the only work that employs a text summarizer based well-established semantic analysis to reduce the inverted index and improve the IR performance.
2. An efficient model for semantic-based text extraction: the thesis presents a semantic-based text analysis model that established a framework on how to combine the traditional statistical

approaches based on tf (term frequency in the text) and idf (the number of documents that contain the term) with semantic analysis approaches in the text mining fields in such a way that improves the performance and reduces the complexity. In this investigation, traditional statistical techniques such as the VSM model and the Jaccard model failed to produce the desirable extracts. In [(Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Yang, Bu, & Xia, 2012), (Wang & Ma, 2013), (Froud, Lachkar, & Ouatik, 2013) , (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015) , (Babar & Patil, 2015), (Ngoc & Tran, 2018)] the authors proved that the processing of the text using the latent semantic analysis increases the precision and recall in the text extraction field, but the time penalty was high, which prohibited the use of the LSA for huge documents.. Therefore, the proposed model draws a map on which kind of text analysis should be used for each part of the text. In our method, we minimize the number of times we should call the LSA procedure, and we perform matrix reduction of the original matrix that represents the text before calling the LSA procedure; this increases the acceptability of using the latent semantic process for large documents.

3. Significant employment of the sentence centrality: many researchers employed the idea of the centrality feature (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (AbdelFattah & Ren, 2009), (Ferreira, et al., 2013). And, we can not pretend that our work is the only work that investigates the centrality of the sentences, but in the mentioned references, the researcher used simple techniques to measure the centrality and used it with a combination of features, which leads to a mysterious though about its effect. Thus, we consider our work as the only work that uses the centrality as the only distinguishing feature and the only work that uses efficient semantic analysis to measure the sentence centrality.

4. Robust evaluation strategy: automatic evaluation tools for assessing the quality of the automatically generated extracts assume that the generated extracts have fixed-sized, Such tools give a significant indication of the extraction precision, but with variable-sized automatic extracts that are necessary for other fields of text processing such as the information retrieval, these tools give misleading evaluation because it evaluates extracts of different sizes. We propose the containment evaluation that measures the percent of the complete sentences that are shared between the reference and the automatic summaries and takes into account the condensation rate.

5. Efficient and accurate semantic-based synonym extraction. The NBDV is proposed to enhance the synonyms extraction through the amendment of the current tf.idf weighting schemes used in the models that adapted the VSM to be applicable to work in synonyms extraction such as the skip-gram model or the Continuous Bag-of-Words model (Mikolov, Chen, Corrado, & Dean, 2013). In SG and CBoW models, the word vector of the word w holds the weights of every word with respect to w in the corpus. If we have t words in the whole corpus, this means that we need $O(t^2)$ time complexity to construct the words-words similarity matrix. Then, the VSM computes the cosine similarity between the terms that also takes $O(t)$. The total complexity of the VSM is $O(t) +$

$O(t^2)$ that implies $O(t^2)$. The cost is high, especially if we want to use the synonyms extraction system in real-time to support the other text mining applications. Therefore, enhancing the weighting scheme is an important issue in synonyms extraction methods. The Objective of the NBDV method is to obtain fast and accurate synonyms extraction by using an enhanced weighting scheme (OWS) in the weighting phase of the vector-space based synonyms extraction. In the OWS, the words that should be processed are only the words that have semantic relation with the word that we want to find its synonyms. In each run, the similarities are computed between the nouns that share distinctive verbs, and the set of distinctive verb of the noun is chosen by considering important factors, (1) the number of times the verb appeared with the noun, (2) the number of nouns the verb appeared-with in the corpus, and (3) the average distance between the verb and the noun in each occurrence of verb and noun together. These factors are necessary to measure the uniqueness of the verb with respect to a specific set of nouns.

1.8 Chapter Summary

This chapter presented the thesis objectives and contributions in the IR and NLP fields. The introduction chapter briefly introduced the text extraction models that will be used to produce effective and informative extraction and to boost the IR performance through the production of informative and reduced inverted index.

The chapter pinpointed the measure differences between the proposed solution in the thesis and the research efforts on the same track. Therefore, it was necessary to perform a deep literature review study that investigates the effect of the different text mining strategies on the relevancy and performance of the IR systems. The next chapter surveys the latest techniques and models that have been developed in the field of automatic text extraction and shows how these models affected the relevancy of the IR system.

1.9 Thesis Organization

The thesis starts by presenting the importance of this research in chapter 1 that includes the research objectives and contributions. Chapter 2 reviews the recent publications that investigated the enhancement of the AIR systems. The review includes the achieved enhancement through Stemming (section 2.2.1), Indexing (section 2.2.2), Query Expansion (section 2.2.3), Automatic Text Summarization (section 2.2.4), Text Translation (section 2.2.5), and Named Entity Recognition (section 2.2.6). Also, chapter 2 reviews the recent publications in the automatic text extraction (section 2.3) and the automatic synonyms extraction (section 2.4).

Chapter 3 describes the methods that are used to extract the salient text segments (MLS text extraction in section 3.3) and to extract the query terms synonyms (NBDV synonyms extraction in section 3.4). Section 3.5 presents the IR model that matched the query and documents terms, and section 3.6 discusses the merits and deficiencies of the methods that are described in section 3.3 and 3.4.

Chapter 4 depicts the experiments used to test the methods that are proposed and designed in Chapter 3. The experiments' environment is described in section 4.2 that includes the data sets (section 4.2.1) and the experiment setting (section 4.2.2). The collected results from the experiments are depicted in section 4.3. These results include the collected results from the MLS text extraction (section 4.3.1), the collected results from the NBDV synonyms extraction (section 4.3.2), and the collected results from the IR experiments (section 4.3.3).

The collected results in Chapter 4 are evaluated in Chapter 5. Sections 5.2 and 5.3 give the intrinsic evaluation of the automatic text extraction method (MLS) and the automatic synonyms extraction method (NBDV). Section 5.4 evaluates the relevancy and efficiency of the results in section 4.3.3 (the results of employing the MLS and NBDV in the Arabic IR system).

Chapter 6 presents the final conclusions and future works. Section 6.1 draws the final conclusions of the intrinsic evaluation that appears in sections 5.2 and 5.3 and presents the limitations that appeared in the evaluation phase. Section 6.2 draws the final conclusions of the extrinsic evaluation that appears in section 5.4 and presents the limitations that appeared in the evaluation phase. In section 6.3, we chose to revisit the objectives (section 1.6) of the research to show how we achieved them, and in section 6.4, we provide the evidence from our results that supports the contributions proposed in section 1.7. Finally, section 6.5 shows the future plans of the methods described in sections 3.3 and 3.4 and depicted the possible improvements.

CHAPTER 2 LITERATURE REVIEW

The presented research integrates the VSM model with a text summarization model to reduce the inverted index size and with a synonyms extraction model to expand the user query. Therefore, the literature review chapter includes three main parts: section 2.1 and 2.2 presents background about IR and reviews the different methods and approaches that were proposed in the literature to enhance the relevancy of the Arabic IR systems. Section 2.3 traces the development in the field of automatic text summarization, and section 2.4 explains the different approaches used to extract the synonyms.

2.1 IR Preprocessing and Matching Operations

The general goal of any information retrieval system is to retrieve the set of documents that satisfy the searcher's information need (Schütze, Christopher, & Prabhakar, Introduction to information retrieval, 2008). The searcher information need is a piece of information in a specific field of knowledge. The user transfers his/her information needs to a set of query terms. The query terms and the terms of the documents are mapped to the index file, and the IR system inspects the index and makes the required matching between the query and the terms of the documents.

Figure 1.2 explains the basic operations that are embedded in the IR system, and we explained two of them (the indexing and the query expansion) in the introduction chapter because those operations are the primary concern of this research and we pinpointed through them the enhancement proposed in this research. But, the indexing operation is preceded by the preprocessing operations that normalize the text under certain criteria and make it more beneficial during the matching process.

2.1.1 IR Preprocessing

Before any searching process is initiated, the information retrieval system needs to map the text that represents the collection of documents to a set of index terms. A complicated process called document preprocessing is required as an initial step to make the text ready for indexing (see Figure 2.1). Before this process, the computer sees the

text as a collection of characters that contains letters, digits, word separators, punctuation marks, some special characters (- , ; , &, …). Therefore, we need to restructure the text such that the computer can deal with it efficiently.

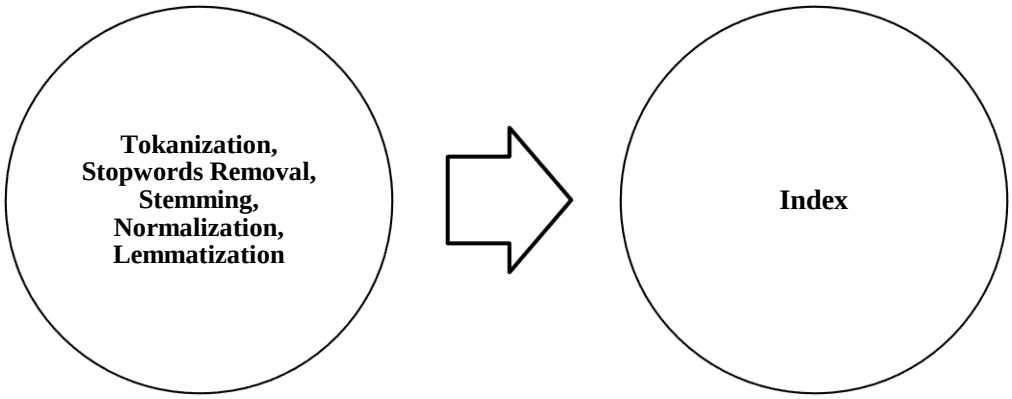


Figure 2.1 IR Preprocessing Operations

Commonly, the preprocessing stage includes the following stages:

Tokenization: It's a Lexical analysis process that aims to separate the text in to set of words (Tokens) using word separators such as space or newline (William & Ricardo, 1992).

Stopwords removal: usually, it's the second stage that aims to remove Stopwords from the collection. Stopwords are found frequently in the text and do not have any meaning by themselves. According to (Al-Shalabi, Kanaan, Jaam, Hasnah, & Hilat, 2004), the list of Stopwords for English exceeded 400, and it includes articles (the, a, an), prepositions (of, in, to,…), pronouns (he, she, you,…), conjunctions (not, and, or, …), adverbs(here, now, very,…), and others. Elimination of Stopwords improve the IR system efficiency as well as effectiveness because it reduces the index size, and the index will not contain meaningless data that might confuse the IR system (Al-Shalabi R. , Kanaan, Jaam, Hasnah, & Hilat, 2004).

Normalization: is the process of mapping the terms of the document and the query terms to one standard form. For example in Arabic we have four variations for the letter " أ " that include " أ "," آ "," إ ", " ا " and the normalization process unified them to one form " ا ", in the same way, the letters " ة " and " ه " are normalized to " ه ".

Stemming*:* after determining the boundaries for each word, it's important to find the stem or root for each word. Many research publications showed the importance of stemming in the IR system (Aljlayl & Ophir, 2002), (Abu-Salem & Philip, English-Arabic cross-language information retrieval based on parallel documents., 2006). The stem represents the basis of the word after removing all the prefixes and suffixes. (Duwairi, Al-Refai, & Khasawneh, 2007), (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, 2009)

Lemmatization*:* it is the process of removing the inflectional ending of a word by using the morphological analysis process. The lemma -which is the output of lemmatization- represents the base or dictionary form of the word, whereas the stem represents the common form between groups of words. For example, the words produce and production have “produc” as a common form and “produce” as a lemma. This means that stemming may return a misspelled word, but a common string between groups of semantically related words. Also, lemmatization employs thesaurus to find synonyms of the words. For example, car and automobile will be mapped to the car as a lemma (Christopher, Prabhakar, & Hinrich, 2009).

2.1.2 IR Models

The index represents the preprocessing and indexing stages output, and at the same time, it's the input to the information retrieval model. The IR model uses the index and the user query to make the matching process that produces a set of documents relevant to the user query. Figure 1.2 depicts the use and position of the IR models. In the IR field, three models are widely used and tested; the Boolean model, the Vector Space Model, and the Probabilistic model.

2.1.2.1 Boolean Model

The Boolean model or the exact match model returns a set of documents that match a specific set of terms connected with logical operators OR, AND, and NOT (Baeza-Yates & Ribeiro, 2011). This means that the user query should be written using Boolean expressions and any document satisfies the expression will be retrieved. Conversely, the document that does not exactly match the expression will be excluded. The implementation of such a model is straightforward, but it has the important limitations shown in Table 2.1.

2.1.2.2 Vector Space Model (VSM)

An algebraic model for matching documents and queries (Baeza-Yates & Ribeiro, 2011). The documents and queries are depicted as vectors in multidimensional space. The components of each vector are a set of terms' weights that reflect the importance of these terms in the document.

$$\vec{\mathbf{d}_j} = (\mathbf{w_{1,j}}, \mathbf{w_{2,j}}, \mathbf{w_{3,j}}, \dots, \mathbf{w_{n,j}})$$

$$\vec{\mathbf{Q}} = (\mathbf{w_{1,q}}, \mathbf{w_{2,q}}, \mathbf{w_{3,q}}, \dots, \mathbf{w_{n,q}})$$

where $\vec{d_j}$: the vector of document j in the collection, $w_{1,j}$: is the weight of the term 1 in document j, $q_{1,j}$: is the weight of the term 1 in the query.

An important issue that should be considering when we talk about VSM is the weighting scheme. No standard weighting scheme is found (Baeza-Yates & Ribeiro, 2011), but the best-known weighting scheme proposed by Salton in (Salton, Wong, & Chungshu, A vector space model for automatic indexing, 1975), it is called the tf.idf weighting scheme, where tf is the frequency of term i and idf is the number of documents that contain i,

$$\boldsymbol{w_{t,d}} = (\mathbf{1} + \boldsymbol{log}\, \boldsymbol{f_{t,d}})(\boldsymbol{log}\, \frac{\boldsymbol{N}}{\boldsymbol{dft}})$$

Where $w_{t,d}$ is the weight of the term t in text d, $f_{t,d}$ is the frequency of the term t in text d, dft is the number of text segment contains t, N is the number of text segments in the corpus; text segment could be document or query.

In the tf.idf weighting scheme, the terms that frequently appear in a certain document, and distributed over a few numbers of documents take more weights than the terms that appear in every document. Thus, the stopwords and the general nouns and verbs, which appear everywhere in the text and do not represent concepts or topics, gain insignificant weights.

After computing the weights and preparing the documents' vectors, VSM calculates the similarity between each document and the user query by computing the cosine of the angle between the vectors that represent them (Schütze, Christopher, & Prabhakar, Introduction to information retrieval, 2008).

$$\boldsymbol{sim}(\vec{\boldsymbol{d_j}}, \vec{\boldsymbol{Q}}) = \boldsymbol{cos}(\vec{\boldsymbol{d_j}}, \vec{\boldsymbol{Q}}) = \frac{\vec{\boldsymbol{d_j}}.\vec{\boldsymbol{Q}}}{|\vec{\boldsymbol{d_j}}|.|\vec{\boldsymbol{Q}}|} = \frac{\sum_{i=1}^{t} \boldsymbol{wd_{j_i}}.\boldsymbol{wQ_i}}{\sqrt{\sum_{i=1}^{t} \boldsymbol{wd_{j_i}}^2} \cdot \sqrt{\sum_{i=1}^{t} \boldsymbol{wQ_i}^2}}$$

Where $\vec{d_j}$ is the vector of document j, $\vec{Q}$ is the vector text query Q, wd_{j_i} is the weight of the term i in d_j, wQ_i is the weight of the term i in Q, t is the number of terms in the whole corpus

The VSM is the most widely used model in information retrieval and natural language processing (Dai, Diao, & Zhou, 2005), (Singh & Dwivedi, 2013), (Luo, Yinglin, Xue, & Zhenda, 2018). The retrieved set of documents obtained by the VSM model is ranked according to their cosine similarity value, and the model allows the partial match. However, this model assumes that the terms are independent, which sometimes does not reflect the real situation, for example, the term " network " normally appears with the term " computer " in the same document and the appearance of " network " strongly recommended the appearance of " computer ".

2.1.2.3 Probabilistic Model

The probabilistic model assumes the existence of a typical set of relevant documents for a specific query and the query describes the properties of this set. This model uses the probabilistic framework and ranks the retrieved set according to their relevance probability to the query (Christopher, Prabhakar, & Hinrich, 2009). The probabilistic model estimates (not exact value) the probability that a document d_j belongs to the typical set of documents with respect to query q. Table 2.1 gives a comparison of the basic IR models with their strengths and limitations.

2.1.2.4 IR Modern Models

Both the VSM model and the probabilistic model assume that the terms are independent, a new modern modal called Set-based Model (Baeza & Ribeiro, 1999) was the first model employs terms' mutual dependencies to obtain more accurate results.

Boolean Model is the weakest model with no partial match and no ranking for the retrieved set. The Extended Boolean Model (William & Ricardo, 1992) handles the partial matching and computes weights for terms using characteristics of the vector model and Boolean algebra.

Another model based on Fuzzy logic and extended Boolean model is the Fuzzy Model. Mixed Min and Max, PAICE, and P-NORM are the most effective variations of the Fuzzy model (Bo-Yeong, Dae-Won, & Hae-Jung, 2005).
Latent Semantic Indexing (LSI) (Baeza & Ribeiro, 1999) was proposed in the field of Information Retrieval to address two main problems with the vector space model, synonyms and polysemy. LSI employs Singular Value Decomposition (SVD) to gain a better understanding of the text being processed. SVD is an algebraic procedure that can be used to explore the relationships among terms that will support any operation applied to a natural language text. (More description can be found in the Latent Semantic Section 2.3.2)

Table 2.1 IR Classical Models

Model	Document representation	Query Representation	Strengths	Limitations
Boolean	Logical conjunction of keywords (not weighted)	any Boolean expression of keywords	• Easy to implement. • Neat formalism	• Binary Retrieval base and no partial matching. • Unranked retrieval list. • Formulation of the Boolean query is not easy for regular users
Vector Space Model	Document and query are represented as vectors in multidimensional space		• It has a mathematical foundation. • The weights reflect the importance of the term. • It shows sufficient effectiveness in information retrieval.	• no formal method for weight computations. • Terms independence assumption. • Lack of finding relationships between terms
Probabilistic Model	Document and query are represented as sets of distinct terms.		• Documents are ranked in decreasing order of their probability being relevant	• Required an initial guess of the relevant documents • The number of occurrences of terms does not support their weight. • Heavy and complicated computations are required

In the introduction section, we specify that the goal of the research is to improve the performance of the IR system by benefiting from the ATS methods that use Multi-Layers semantic analysis. The literature review contains three main sections that are related to the major contributions that are mentioned in the introduction chapter. The first section reviews the developed and experimented techniques to enhance the relevancy of the Arabic IR systems. The researchers of the AIR performed syntactic and semantic text analysis processes to extract significant information from the text, which participated in the increase of the relevancy of their systems. So in section 2.2, we collect, analyze, and abstract their main results to offer a summary that can be used as a base in building an Arabic IR system. Section 2.3 reviews the literature of the Automatic Text Extraction after introducing a general background of ATS. The ATE is the kind of ATS used in this research, and we reviewed the approaches, methods, and achievements obtained in this field. Section 2.4 surveys the models used to extract the words' synonyms with their approaches and results.

2.2 AIR Researchers Efforts

Numerous techniques have been developed to improve the relevancy of AIR systems. In one hand, some researchers attempted to enhance the IR system by improving the IR system components such as:[5]

1. Building high accurate stemmer suits the special features of Arabic language (Al-Shalabi, Kannan, Hilat, Ababneh, & Al-Zubi, 2005), (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced algorithm for extracting the root of Arabic words, 2009), (Khoja, 2012), (Larkey, Ballesteros, & Connell, Light Stemming for Arabic Information Retrieval, 2007).

2. Improving the index structure in a way that reduces the time required to obtain the relevant documents (Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval, 2014), (Ghassan, Riyad, & Sawalha, 2005).

3. Enhancing the user query by inserting new semantically related terms to the query terms. (Abbache, Barigou, Belkredim, & Belalem, 2016), (Abdelali, Cowie, & Hamdy, 2007), (Atwan, Mohd, Rashaideh, & Kanaan, 2016) , (Hanandeh, 2013), (Mallat, Anis, Emna, & Mounir, 2013) , (Shaalan, Al-Sheikh, & Farhad, 2012) , (Wedyan, Alhadidi, & Alrabea, 2012).

On the other hand, some of the surveyed researchers attempt to benefit from the NLP tasks. For instance:

1. Text summarization can reduce the document size, which results in reducing the index size and accelerating the retrieval process.

2. Named Entity Recognition (NER) determines the entities that are mentioned in the documents being searched.

3. Machine Translation (MT) and Machine Readable Dictionary (MRD) may use to translate the query and initiate the search over other languages.

In the next subsections, the recent research that studied the effect of the NLP tasks on the relevancy and efficiency of the AIR has been investigated.

[5] Parts of this section and its subsections are mentioned in the first paper of the "**Publications Arising from This Thesis**" section

2.2.1 Stemming Impact

Stemming is the process of stripping derivational and inflectional affixes of a word. Stemming ensures that all variants of the word will be treated in a similar way by the NLP or the IR systems (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, 2009).

For the Arabic language, we can abstract four strategies used in literature to stem Arabic words:

1. Rule-based Affix removal: affixes stored in the database dictionary. The dictionary holds Arabic language affixes such as "ال", "ون", "ها", "لل", "ين", "ان", "ات","بال","ف". Simple control statements remove those affixes out from the word. This strategy used in (Al-Kabi, Towards Improving Khoja Rule-Based Arabic Stemmer, 2013), (Al-Shalabi & Kanaan, Stemmer Algorithm for Arabic Words Based on Excessive Letter Locations, 2007), (Khoja, 2012), (Larkey, Ballesteros, & Connell, Light Stemming for Arabic Information Retrieval, 2007).

2. Statistical based Affix removal: In which statistical calculations determine the most important part of the processed word. This strategy used in (Al-Shalabi, Kannan, Hilat, Ababneh, & Al-Zubi, 2005), (Hafer & Weiss, 1974), (Hmeidi, Alshalabi, Al-Taani, Najadat, & Al-Hazimah, 2010).

3. Pattern matching strategy which is mainly a morphological pattern matching process to extract three, four, five, or even six letters Arabic roots (Al-Kabi, Towards Improving Khoja Rule-Based Arabic Stemmer, 2013), (Al-Kabi, Kazakzeh, Abu Ata, Al-Rababah, & Alsmadi, 2015), (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced algorithm for extracting the root of Arabic words, 2009), and (Khoja, 2012).

4. Dictionary-based stemming in which the stems of words are stored in a lexicon. This strategy used in (Alhanini & Aziz, 2011). Mainly, Arabic IR researchers mixed more than one strategy to enhance the accuracy level. (Accuracy level equals correct stems divided by the number of inputted words).

Table 2.2 summarizes the main stemmers developed for the Arabic language with their accuracy level and approaches.

The important matter in this context is to measure the impact of stemming On AIR relevancy results. In (Darwish & Oard, CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval, 2002), three stemmers were tested, Al-Stem[6], UMass, and Modified UMass stemmers. The authors used TREC2001 &

[6] www.glue.umd.edu/~kareem/research

2002 data, and the obtained MAP values were 32%, 32%, and 33%, respectively. Light 10 stemmer in (Larkey, Ballesteros, & Connell, Light Stemming for Arabic Information Retrieval, 2007) showed significant improvement and achieved 41% AP. Bessou and Touahria in (Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval., 2014) obtained 57% AP and 69% AR when they tested the effects of their stemming on AIR relevancy. They achieved 15% improvement in AP and 28% in AR comparing with no stemming IR system.

Aljlayl and Frieder in (Aljlayl & Frieder, On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, 2002) used AP to compare the relevancy of three AIR system, the first one called surface-based retrieval in which no stemming where performed, the second one is root-based retrieval in which they used aggressive algorithm to extract the words' roots, and the final system used light stemmer to remove the prefixes and suffixes of the words. They showed that the AP improved by 43.2% using root based stemming and by 71.3% using light stemming (surface-based stemming AP=25%, root-base stemming AP = 36%, and light stemming based AP=43%).

Chen and Gey in (Chen & Gey, 2002) studied the impact of light stemming and Machine translation based stemming on AIR. The authors used precision and recall relevancy measures to show that the light stemmer outperformed the MT-based stemmer when those stemmers were used as a stemming tool (IR system with MT-based stemmer: the R =83% and P=33%, IR system with light stemmer: R=84% and P=37%).

Stemming accuracy dominated a large space of interest in AIR research. As shown in Table 2.2, the statistical approaches, the rule-based approaches, and the pattern matching approaches succeed to obtain 95% level of accuracy. But, to be more accurate, we computed the Average accuracy achieved by each approach using our references as the sample, and the results are presented in Figure 2.2.

The Positive Effect of stemming appears clearly in all surveyed publications that studied the effect of stemming on AIR. Both stem and root based retrieval returned more relevant documents comparing with full-word based retrieval. For example, in the four publications: (AbdulJaleel & Larkey, 2003), (Aljlayl & Frieder, On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, 2002), (Al-Kharashi & Martha, Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System, 1994), (Hmeidi, Kanaan, & Martha,

Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. , 1997), the AP was boosted with percentages of improvements 25%, 18%, 27%, 8% respectively.

Table 2.2 Stemming Accuracy Obtained from the Surveyed Publications

Ref #	Year	Data collection	Accuracy Level	Stemming approach
(Khoja, 2012)	2001	50,000 words	71%	Rule-based strategy with pattern matching
(Al-Kabi, Kazakzeh, Abu Ata, Al-Rababah, & Alsmadi, 2015)	2015	6081 words	75%	Rule-based strategy with pattern matching
(Al-Kabi, Towards Improving Khoja Rule-Based Arabic Stemmer, 2013)	2013	6000 words	76%	Improvement of Khoja stemmer
(Al-Shalabi, Kannan, Hilat, Ababneh, & Al-Zubi, 2005)	2005	2000 words	80%	Statistical: Successor variety stemming
(Yaseen & Hmeidi, 2014)	2014	1000 documents	83.9%	Pattern matching strategy
(Hmeidi, Alshalabi, Al-Taani, Najadat, & Al-Hazimah, 2010)	2010	---	89.6%	Statistical based
(AL-Omari & AbuAta, 2014)	2014	6225 words	92%	Statistical based
(Al-Shalabi & Kanaan, Stemmer Algorithm for Arabic Words Based on Excessive Letter Locations, 2007)	2007	1000 word	95%	Rule-based Affix removal strategy
(Ghawanmeh, 2005)	2005	----	95%	Pattern matching strategy
(Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced algorithm for extracting the root of Arabic words, 2009)	2009	15180 words	95%	Rule-based strategy with pattern matching
(Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval., 2014)	2014	59548 words	96%	Pattern matching strategy
(Darwish, Building a Shallow Morphological Analyzer in One Day, 2002)	2002	9606 words	96%	Statistical based
(Alhanini & Aziz, 2011)	2011	----	96.29%	Dictionary-based with Rule-based removal
(Lee, Paining, Roukos, Emam, & Hassan, 2003)	2003	28449 words	97%	Statistical based

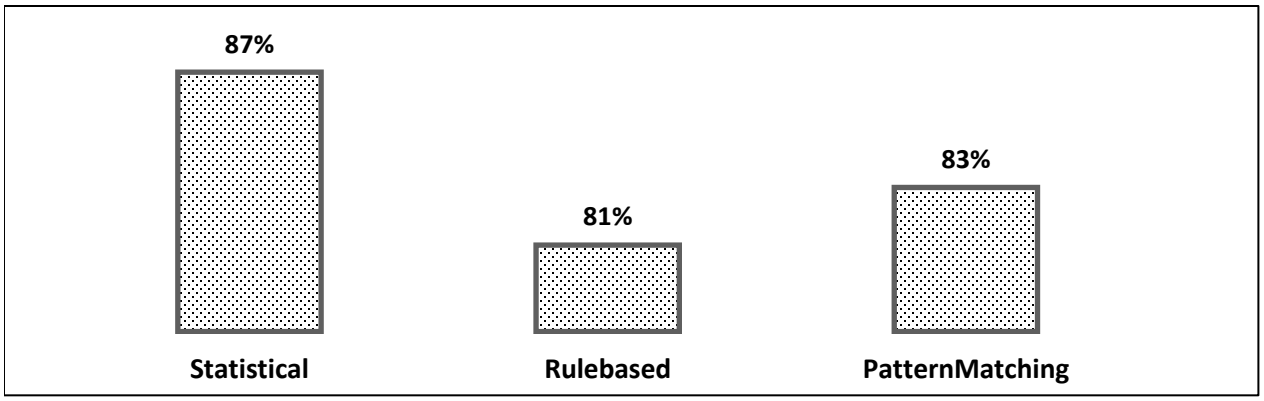


Figure 2.2 The Average Accuracy of Different Stemming Approaches

2.2.2 Indexing Impact

The index is a data structure that represents the documents' contents as a list of weighted terms in the IR systems. The index is inspected against the user query to find which terms inside the index best match the user query. Therefore, a well-established index is necessary to optimize speed and performance (Mansour, Haraty, Daher, & Houri, 2008).

The Index can be a single word (SW), multi-words (MW), or semantic index that relates the semantically related terms (Abderrahim, Mohammed, & Chikh, 2013). The single word entry in the index might be a complete word, stem, or root (represents the base or dictionary form of the word without prefixes, suffixes, and infixes) (Aljlayl & Ophir, 2002), (Bessou & Mohamed, n Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval, 2014), (Al-Kharashi & Martha, Comparing words, stems, and roots as index terms in an Arabic information retrieval system, 1994). Phrase indexing was also investigated. The phrases improve the semantic side of the index and characterize the document contents more effectively than the single word terms (Boulaknadel, Daille, & Driss, 2008). Semantic indexing relays on word senses disambiguation and tries to find the correct sense of the word among different senses. Researchers have examined different strategies to identify suitable indexing strategies to improve the precision measure.

Since 1975, Salton began his investigation about the importance of indexing in the field of IR. In the beginning, the indexing process was performed on manual bases that are mainly expensive, time-consuming, inaccurate, and harder to maintain and update (Salton & McGill, Index construction, 1983). Salton emphasized on term frequency

as a base to select index terms. He neglected the most and least frequent terms, and he indexed midrange terms. For the Arabic Language, the authors of (Hmeidi, Kanaan, & Martha, Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. , 1997) experimented the use of term frequency to select index terms for Arabic language and they indexed the terms that have frequencies greater than one and less than 261. They built an information retrieval system with manual and automatic index and conducted a series of experiments using full word, stem, and root as index terms. The results showed that the automatic indexing is comparable to manual indexing. Also, it is cheaper and faster.

Table 2.3 summarizes the main findings with the IR relevancy measurement calculations and the type of indexing of some of the surveyed publications. The most recent publications that investigated the index structure and the data to be indexed in AIR literature were reviewed in this review. SingleWord-full index, SingleWord-stem index, SingleWord-root index, SingleWord-ngarm index, MultiWord index, and semantic index are employed either manual or automatic. Single-term indexes dominated a large number of those researches with AP swung between 20% and 90%. On the other hand, the semantic index surpassed the regular key terms index. In general, the main findings related to indexing strategies can be summarized in the following points:

1. In (Abu-Salem & Philip, English-Arabic Cross-Language Information Retrieval Based on Parallel Documents, 2006), (Al-Kharashi & Martha, Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System, 1994), (Hmeidi, Kanaan, & Martha, Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. , 1997), (Mansour, Haraty, Daher, & Houri, 2008) the use of root as index term gave the best relevancy results comparing with stem or full word term index. See Figure 2.3.

2. In our survey, we note that the choice between MultiWord indexing and SingleWord indexing tends toward the latter. In SingleWord index: the AP values fluctuated between 20% in (Hmeidi, Kanaan, & Martha, Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. , 1997) and 90% in (Hammouda & Almarimi, 2010). In MultiWord index: the maximum value of AP appeared in our survey for Arabic IR systems with phrases indexing was 37% found in (Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval, 2014).

3. The choice between key terms index and semantic index tends toward the latter. The improvement of MAP values using semantic index over key term index reached 15% in (Abderrahim, Mohammed, & Mohammed, 2016) and 21% in (Abderrahim, Mohammed, & Chikh, 2013).

Table 2.3 References Addressed the Indexing Impact with their Relevancy Assessment

Ref	Index type, Main Finding, and IR Measurement SW: Single Word, MW: Multi Words
(Abderrahim, Mohammed, & Mohammed, 2016)	AP= 39% (key terms index), AP = 54% (semantic index, without disambiguation words resolve), AP=60% (semantic index, with disambiguation words resolve).
(Abderrahim, Mohammed, & Chikh, 2013)	Building semantic index. MAP = 39% (key-terms index), MAP = 60% (semantic index)
(Abu-Salem & Philip, English-Arabic Cross-Language Information Retrieval Based on Parallel Documents, 2006)	SW terms index. MAP=54% (Roots), MAP=42% (Stems), MAP=29% (Full word)
(Aljlayl & Frieder, On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, 2002)	SW terms, Light stemming AP = 43% (Stems), AP = 36% (Roots), AP = 25% (Full word)
, (Al-Kharashi & Martha, Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System, 1994)	SW terms index, MAP=66% (Roots), MAP=58 % (Stems), MAP=39% (Full word)
(Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval., 2014)	SW terms index, pattern matching stemming algorithm, AP=57% (stem), AP=43% (Full word)
(Bessou & Touahria, an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval, 2014)	AP using MW is greater than the AP using SWT by 5.8%. (AP value = 31.9%)
(El-Beltagy, Rafea, & ., 2009)	Use KP-Miner system to extract MW index entries. (AP=19%, AR=43%)
(Elshishtawy & Al-sammak, 2009)	Use statistical measures with the linguistic knowledge to extract MW index entries. P = 65%, R = 40%
(Harrag, Aboubekeur, & Eyas, 2008)	SW stem terms index, VSM as IR matching Model. AP = 66%(Stem), AR = 80% (Stem)
(Hmeidi, Kanaan, & Martha, Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. , 1997)	SW terms index AP = 28.4% (Root), AP=26.8% (Stem), AP=19.8% (Full word)
(Ghassan, Riyad, & Sawalha, 2005)	Nouns as index terms. The AP/AR declined by less than 1%, and the index size shrinks by 45%.
(Mahmoud, Sanan, & Zreik, 2011)	SW terms, experimented n-gram as a source of indexing. R = 41.3% (Stem), P = 32.81%. (Stem)
(Mansour, Haraty, Daher, & Houri, 2008)	SW term index, VSM as IR matching Model. AP=64%(Root), AR=46%(Stem)

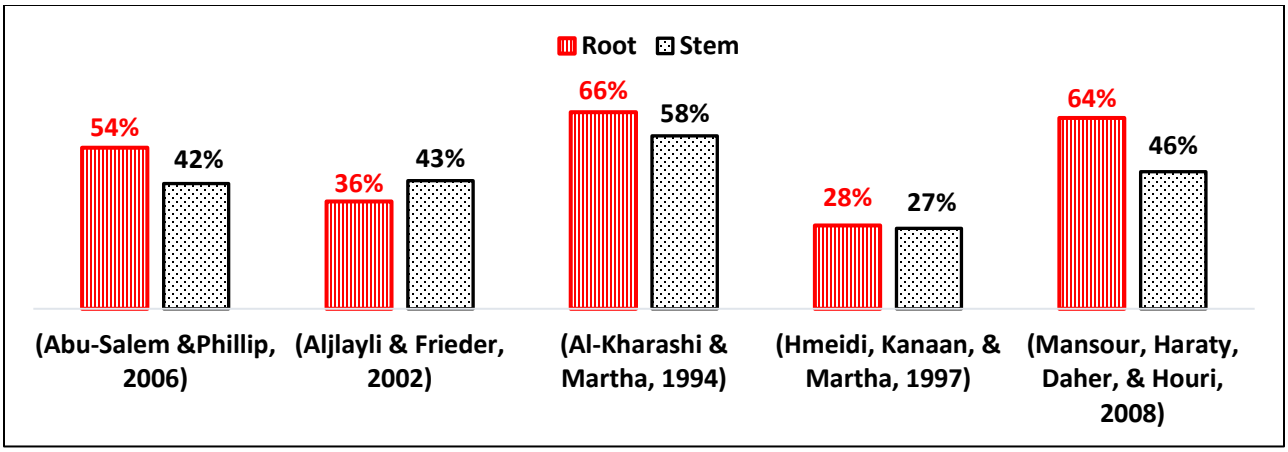


Figure 2.3 AP of the IR Systems that Appeared in Table 3 and Used Root or Stem as Index Entry.

2.2.3 Query Expansion Impact

QE is an IR technique used to improve the relevancy of IR systems through the amendment of the query key terms. Normally, the user query is short and contains few words. And, the user's lack of knowledge of the subject being searched makes the user selects inadequate or out of domain keywords. This technique assumes that the user query is the problem in retrieving irrelevant documents and the relevancy can be improved by either adding new words to the query or by replacing the query words with new words.

Two main methods have been employed to expand users' queries; the first one uses linguistic knowledge to add synonyms to the terms used in the query and drive new semantically related terms to the query terms. The second method uses user feedback to enhance the query. Another categorization of query expansion methods is Global or Local. Global methods perform the expansion by hiring external resources such as lexicon thesaurus or stemming algorithm and Local methods use the data collected from the first run of the IR system to expand the query. The surveyed publications that measure the impact of QE appears in Table 2.4.

The researchers in Arabic information retrieval investigated the problem of expansion from different perspectives: 1) Linguistic analysis with relevance feedback expansion, 2) Automatic and Interactive feedback expansion, 3) Local and global thesaurus based expansion, 4) Local linguistic or statistical analysis expansion, and 5) Local semantic analysis based on statistical calculation or global semantic analysis based on Arabic WordNet.

Our survey showed a positive impact of QE, and this appears clearly in all publications examined. The recall was improved, and the percentage of improvement in MAP lies between 1% in (Hanandeh, 2013) and 14% in (Wedyan, Alhadidi, & Alrabea, 2012) which gives a strong indication that the query expansion recalled more relevant documents and discarded some of the irrelevant ones. Another important note related to precision measure which showed enhanced (Abdelali, Cowie, & Hamdy, 2007), (Atwan, Mohd, Rashaideh, & Kanaan, 2016), (Mallat, Anis, Emna, & Mounir, 2013), (Wedyan, Alhadidi, & Alrabea, 2012) or stable trend (Abbache, Barigou, Belkredim, & Belalem, 2016), (Hanandeh, 2013). See Figure 2.4.

Table 2.4 References Addressed the QE Impact with their Relevancy Assessment.

Ref	Expansion type and Relevancy measurements Values
(Abbache, Barigou, Belkredim, & Belalem, 2016)	Without QE: R=70%, P=21%, and MAP=37%. Automatic QE (Global, thesaurus): R=91% , P=6%, and MAP= 25%. Interactive QE (local, Relevance feedback): R=82%, P=15%, and MAP=41%.
(Abdelali, Cowie, & Hamdy, 2007)	Without expansion: R = 34%, P = 6%. With Local expansion : R=74%, P=12%
(Atwan, Mohd, Rashaideh, & Kanaan, 2016)	Baseline retrieval: R = 47%, MAP = 16%, f_score = 24%. QE using semantic WordNet and semantic similarity: R= 50%, MAP= 29%, f_score = 37%
(Hanandeh, 2013)	Local, thesaurus based, With expansion: MAP = 56.6%, Without expansion: MAP = 55.5%
(Mallat, Anis, Emna, & Mounir, 2013)	Before expansion: R = 31%, P = 33% After expansion (Local, thesaurus): R = 74%, P = 81%.
(Shaalan, Al-Sheikh, & Farhad, 2012)	Before expansion: R=84%, After expansion(Local, statistical base) : R=91%
(Wedyan, Alhadidi, & Alrabea, 2012)	Without expansion: MAP=34%, With expansion(Global, thesaurus based): MAP=48 %

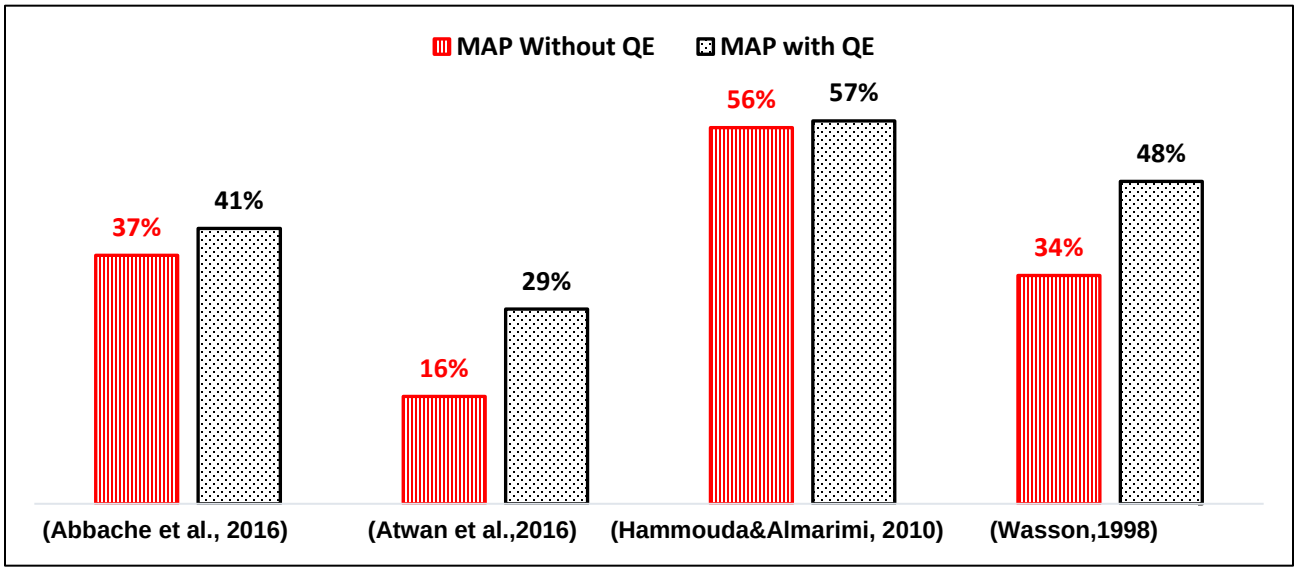


Figure 2.4 the MAP of the Surveyed Publications that addressed the QE

2.2.4 Automatic Text Summarization Impact

ATS is the computer's ability to simulate human being's skills in drawing the main ideas or the key sentences from a particular text. The summary represents important information found in an article. If the summary of the document contains sufficient information, it can be employed in place of the full document itself in the IR systems. We should satisfy the equation that keeps the relevancy of the IR system reasonable and at the same time, reduces the retrieval time.

Firstly, No actual work was found for the employment of ATS in AIR, and we have little work investigated the impact of ATS on the IR performance for the other languages. Brandow et al. in (Brandow, Karl, & Lisa, 1995) obtained a high precision measuring rate when they used a domain-independent automatic summary (extractive summary) based on the traditional tf.idf sentence selection as index source. They compared the results obtained from their extractive summary with another simple summary whose sentences are selected from the first few sentences in the original document (called Lead summary) and also with full-text indexing. They tested the relevancy against three condensation rate 60, 150, 250 words, and they obtained the highest precision value at 250 words summary length. The same conclusion can be driven from Sakai and Sparck-Jones (Sakai & Sparck-Jones, 2001), they employed the same idea of using the summary for indexing, but they studied the impact of Generic summaries with or without Pseudo-relevance feedback. We are considering only the Generic summaries as a source of indexing because we

do not use the Pseudo-relevance feedback because the PRF is a query refinement strategy and out of the scope of this research. Sakai and Sparck-Jones results in (Sakai & Sparck-Jones, 2001) were in line with the results obtained in (Brandow, Karl, & Lisa, 1995), in which a summary-only as source of index obtained the largest precision at the highest condensation rate (see table 5 and 6 in (Sakai & Sparck-Jones, 2001)). Another employment of ATS in IR -especially in Geographical Information Retrieval subfield GIR - done by Perea-Ortega et al. in (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013). They utilized two kinds of summaries, General summary based on word frequency and noun phrases, and Geographic summary that gave more attention to the sentences that contain Geographical entities. The authors in (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013) gave a clear conclusion that the use of statistical single document summarizes as the source of indexing is not significant.

In (Perea-Ortega J. M.-L., 2013), (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001), (Wasson, 1998), the authors used condensation rates of fixed size. Firstly, the CR equals the summary length over the document length, and this parameter could be expressed in terms of a number of words or sentences or as a ratio. Ranald in (Brandow, Karl, & Lisa, 1995) generated summaries containing 60, 150, and 250 words whereas; in (Perea-Ortega J. M.-L., 2013) the condensation rates were 20%, 40%, 60%, and 80% of the original text. Indeed, the use of fixed-sized CR may have a negative impact, and it seems unfeasible especially for IR tasks. In more detail, the richness of information differs from one document to another and a certain document may need 30% CR to capture its entire salient information and another one may need 50%. The IR relevancy measurements of the publications surveyed in this section summarized in Table 2.5.

Table 2.5 References Addressed the ATS Impact with their Relevancy Assessment.

Ref	IR Relevancy Measure and value	Summariztion Technique
(Perea-Ortega J. M.-L., 2013)	No Average of the results has been mentioned, but the authors concluded that the use of statistical single document summarizes as the source of indexing is not significant.	Linguistic and Statistical methods Statistical: based on term frequency, and noun phrases frequency and structure.
(Brandow, Karl, & Lisa, 1995)	AP=37%, AR= 100% full text index AP=45%, AR=59% extractive-summary index	Statistical based on tf.idf
(Sakai & Sparck-Jones, 2001)	AP = 24% At CR = 50% No recall assessment	tf.idf with PRF and Without PRF

2.2.5 Automatic Translation Impact

At first, Machine Translation (MT) is the ability of the computer to translate written or spoken text from one language to another on the sentence level. The IR strategy that uses translation facilities is Cross-Language Information Retrieval (CLIR). CLIR is an information retrieval strategy that returns relevant documents written in several languages. In CLIR, the system normally translates the query and tries to find relevant documents in original and target languages.

In this section, we concentrate on the following points: 1) the use of MT or Machine Readable Dictionary (MRD) as a translation tool, 2) the number of candidate translations generated for each query, and 3) the employment of proper noun transliteration in CLIR.

The queries are translated into another language using either the MT system or the Machine Readable Dictionary. Aljilay et al. (Aljlayl & Frieder, Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation, 2001) held a comparison between two Arabic CLIR systems, the first one used ALKAFI (Basha, 1992) machine translation systems and the second one uses Arabic to English dictionary called Al-Mawrid (Baalbaki, 1988). They found that the retrieval system that uses a bilingual dictionary with a reasonable number of translations for each query term outperformed a retrieval system that uses the MT system to translate the query. Darwish in (Darwish & Oard, CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval, 2002) experimented the use of two Arabic MT systems (Tarjim and AL-Misbar) and one Arabic to English dictionary (Salmone) in IR system. In (Larkey, Allan, Connell, Bolivar, & Wade, 2003), two bilingual dictionaries were used (UMass dictionary & TREC standard probabilistic dictionary).

Abu Salem and Philip in (Abu-Salem & Philip, English-Arabic Cross-Language Information Retrieval Based on Parallel Documents, 2006) used the Sakher MT system to generate a translation matrix for each query. One, two, or three candidate translation(s) for each query term were generated. Their experiment showed that the retrieval performance of CLIR using the MT system outperformed the monolingual retrieval, especially when they used the complete word as an index term, not stem or root. The results of Abo Salem and Philip's experiments are consistent with Hull and Gregory (Hull & Gregory, 1996) who showed that word –by- word translation degraded the retrieval performance by 40% to 60%.

A simple CLIR system can use a Bilingual dictionary to translate the query terms without using a complete MT system. The problem with this kind of dictionary is the out of vocabulary (OOV) words, which are mainly proper nouns for persons, places, and others that are missing in such a dictionary. In (Bellaachia & Ghita, 2008), Bellaachia and Amor-Tijani used English to Arabic dictionary to do the necessary translation and every OOV word was transliterated using statistical techniques followed by n-gram string matching. Before Bellaachia and Amor-Tijani in (Bellaachia & Ghita, 2008), Larkey et al. in (Larkey, AbdulJaleel, & Connell., What's in a name?: Proper names in Arabic cross language information retrieval, 2003) conducted two levels of experiments in this context. Firstly, they found that the MAP jumped from 14% in the case of no name transliteration to 33% in the case of name transliteration. Secondly, they found that as the number of transliterations for the query proper nouns increased, the MAP increased. Abdul Jaleel and Larkey in (AbdulJaleel & Larkey, 2003) experimented a simple technique for statistical n-gram transliteration called selected n-gram. They tested the effectiveness of this technique on Arabic IR. The authors tested the mapping of each name and word to 20 transliterations. The IR relevancy measurements of the publications surveyed in this section were summarized in Table 2.6.

We can conclude that the use of a word by word bilingual dictionary to translate the user query will degrade the retrieval system. Therefore, the researchers improved their results by expanding the query by taking several translations for each query term. For example, in (Abu-Salem & Philip, English-Arabic Cross-Language Information Retrieval Based on Parallel Documents, 2006) the authors took 2 or 3 term translations, and in (Bellaachia & Ghita, 2008) they expanded the query by taking different proper nouns transliterations. On the other hand, the use of Machine translation systems does not give the expected enhancement to the CLIR system. For example, in (Aljlayl & Frieder, Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation, 2001) the AP with MT (17%) was less than the AP of the monolingual retrieval (27%). Normally, using MT to translate the user query will yield a single translation because the MT generates one translation for each query term. As stated by the authors of (Zhou, et al., 2012), the output of MT is a literal mapping, and this ignores the availability of multiple expressions in the target language.

Table 2.6 References Addressed the MT Impact with their Relevancy Assessment.

Ref	IR Relevancy Measures and Values
(AbdulJaleel & Larkey, 2003)	AP=15% baseline, AP=20% with names transliteration, AP=21% with words transliterations
(Abu-Salem & Philip, English-Arabic Cross-Language Information Retrieval Based on Parallel Documents, 2006)	AP= 45% (Two candidate translations), AP= 42% (Three candidate translations)
(Aljlayl & Frieder, Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation, 2001)	AP= 17% (MT as translation tool), AP= 20% (bilingual dictionary translation)
(Bellaachia & Ghita, 2008)	MAP= 46% (without transliteration), MAP= 72% (with transliteration)
(Darwish & Oard, CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval, 2002)	MAP= 33% (two Arabic MT system -Tarjim, AL-Misbar- and one Arabic to English dictionary -Salmone- employed in the IR system
(Larkey, AbdulJaleel, & Connell., What's in a name?: Proper names in Arabic cross language information retrieval, 2003)	MAP=14% without transliteration, MAP=19% with one transliteration MAP=25% with five transliterations, MAP=30% with 20 transliterations
(Larkey, Allan, Connell, Bolivar, & Wade, 2003)	MAP= 40% UMass dictionary & TREC standard probabilistic dictionary

2.2.6 NER & POS Impact

Named Entity Recognition (or NER): It is an information extraction task that aims to identify the entities found in a given text document. The output of NER systems is mainly a list of proper nouns of the entities that are mentioned in the text.

Part of Speech Tagging (POS): It is considered an essential tool for any robust NLP application. It's a morphological analysis process in which we determine the lexical form for each word in the text (Noun, Adjective, Verb ...).

Firstly we want to explain that this section combined Named Entity Recognition (which is NLP task) with POS tagging (which is a morphological analysis process) because both of them have been employed in IR systems in a similar manner. Mainly, POS taggers and NER systems are used in IR to measure the effect of the nouns in the IR systems whether they are common nouns or proper nouns. Unfortunately, the NER task has not taken that much of attention from Arabic IR researchers and a little work can be found in this area.

For English, in (Guo, Xu, Cheng, & Li, 2009), the authors addressed the existence of Named entities in the user's queries, and they found that 71% of queries contained Named Entities. In 1997, Thomson in (Thompson & Dozier,

1997) investigated the effect of NER systems in IR systems. They claimed that the effective NER system would support the IR processes and improve their outcomes. The Information retrieval system developed by Thompson and Dozier benefited from the NER system and obtained 83.9% average precision comparing with baseline IR system, which yielded 74.8%. Another experiment in English language and other languages (Dutch and French) reported in (Kumar, De Beer, Vanthienen, & Moens, 2006), in which four off-the-shelf name entity recognition tools were used to extract proper names.

The output of the NER system is a set of proper nouns. Abdur Chowdhury in (Chowdhury & McCabe, 1998) experienced the use of nouns index instead of the full-text index in Information retrieval. In (Ghassan, Riyad, & Sawalha, 2005), the authors adopted the hypothesis that nouns are the most important part of any text document and could be used as IR discriminator. They analyzed their corpus and found that 55% of the words are nouns. Their experiments showed that the use of nouns as an index term obviates the need for using another part of speech types. The IR relevancy measurements of the publications surveyed in this section were summarized in Table 2.7.

Table 2.7 References Addressed the NER Impact with their Relevancy Assessment.

Ref	IR Relevancy Measure and values	Language
(Chowdhury & McCabe, 1998)	MAP=13% (full text index) , MAP = 12% (proper noun index)	Arabic
(Ghassan, Riyad, & Sawalha, 2005)	MAP=29% (full text index) , MAP = 28% (proper noun index)	Arabic
(Kumar, De Beer, Vanthienen, & Moens, 2006)	P= 97% (with NER system), R= 68% (with NER system),	English, Dutch, French
(Thompson & Dozier, 1997)	AP= 84% (with NER system), AP = 75% (without NER system)	English

We found modest effort spent in this NLP task. In general, the output of the surveyed NER systems showed a positive impact. The nouns generated by the NER system and used as index terms did not degrade the IR system relevancy, and they improve the efficiency. For example, in (Hull & Gregory, 1996) and (Chowdhury & McCabe, 1998), the authors showed that the use of nouns as index terms obviated the use of other types of speech.

2.3 Automatic Text Summarization

As discussed in the introduction chapter, the enhancement proposed in this research depends mainly on building efficient text summarizer. Therefore, it is important to survey the text summarization approaches found in the literature of ATS[7].

2.3.1 Automatic Summaries Classifications

The automatically generated summaries can be classified according to different factors such as the purpose, structure, content, and the type of input stream (Mani, Automatic Summarization., 2001), (Mei & Chen, 2012), (Gambhir & Gupta, 2017). The classifications help to understand the useful type of summary that is necessary for a particular area. The classifications of the summaries include the following:

1. Extracts vs. Abstract

The extracts are produced from the text extraction process in which we copy the salient sentences from the original text without making any change in the copied sentences. Whereas, the abstract is the summary that identifies the salient parts of the text and rewrites and reorders the sentences to produce a summary that resembles the human-generated summary. The abstract is a coherent text that is produced to shorten the time needed to read the newspaper or a long text (Mani, Automatic Summarization., 2001), (Pierre-Etienne & Guy, 2011), (Song, Huang, & Ruan, 2018).

2. Informative vs. Indicative

The indicative summary is used in the search engine to gives the searcher selective parts of the retrieved documents, and according to these parts, the user may discard the document or read the full document and consider it as relevant. Whereas, the informative summary tries to investigate all the salient information to give the reader a complete idea about all the contents of the text (Gambhir & Gupta, 2017).

3. Multi-document vs. Single-document

[7] Parts of this section and its subsections are mentioned in the second paper in the "**Publications Arising from This Thesis**" section.

The summarization of multiple documents in a single summary is called multiple-document summarization. Whereas, the single-document summarization produces a single summary for each document (Mani, Automatic Summarization., 2001).

4. Generic vs. Focused

The focused summary is constraint by predetermined factors such as the document title or the user query. While the generic summary is a miniature version of the original text that contains all the salient parts of the text without considering any initial requirements during the summarization process (Mani, Automatic Summarization., 2001).

5. Words vs. paragraphs

The extraction process either extracts a set of sentences and forms paragraphs or returns a set of words that may represent the named entities in the text or the synonyms of a given word. Distinctive word summaries include Named Entity Recognition, Topic identification (TI), and Synonyms Extraction. NER is an information extraction task that aims to identify the entities that are mentioned in a given text document. The output of NER systems is mainly a list of proper nouns. (2) Topic identification is the process of classifying the documents under a set of predefined topics (Echeverry-Correa, Ferreiros-López, Coucheiro Limeres, Córdoba, & Juan Manuel, 2015). And the synonyms extraction is a text extraction process aimed to find the terms synonyms (Crouch, 1990).

2.3.2 ATS Approaches

The automatic text summarization is being studied since the mid of the previous century (Luhn, The Automatic Creation of Literature Abstracts, 1958), (Baxendale, 1958). Statistical and Linguistic approaches have been used to produce automatic summaries. These approaches investigate certain features found in the text to determine the summary sentences.

2.3.2.1 Statistical Approaches

Statistical approaches give each sentence a numerical score and rank the sentences in the document based on the computed score. The statistical features include the word frequency, the inverse document frequency, the sentence

resemblance to the title, the aggregated similarity, the positive keyword, the negative keyword, the centrality, the Inclusion of name entity, the inclusion of numerical data, the relative sentence length, Bushy path similarity, and others. Table 2.8 shows the features examined in the literature of ATS.

The statistical approaches that are experienced in the field of text summarization include:

- The VSM model based on tf.idf weighting scheme and cosine similarity (El-Haj & Hammo, 2008), (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced Algorithm for Extracting the Root of Arabic Words, 2009), (Kiyoumarsi, 2015)],
- Information retrieval approaches such as the bushy path and aggregate similarity Ferreira, et al., 2013).
- Fuzzy Logic (Babar & Patil, 2015),
- Latent Semantic Analysis: the classical employment of LSA based on SVD matrix decomposition appears in many research efforts (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Yang, Bu, & Xia, 2012), (Wang & Ma, 2013), (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015), and (Babar & Patil, 2015)],
- Neural Networks approaches such as the deep auto-encoder method, the sequence to sequence model, the Feed Forward NN model, and others (Abdel Fattah & Ren, 2008), (AbdelFattah & Ren, 2009), (Yousefi & Hamey, 2017), (Song, Huang, & Ruan, 2018).

The survey in our research shows that no unanimous decision on the ideal feature or combination of features that best describe the text, also there is no unanimous decision on the best scoring equation. Thus, all the surveyed publications represent the experiments of applying the different statistical models on the text summarization and measure the effectiveness of these models in the precision of the automatic summarization.

The features that have been investigated in the ATS field: Ferreira et al. (Ferreira, et al., 2013) found that is: tf, tf.idf, lexical similarity, and sentence length are the best combination of features, whereas Meena and Gopalani (Meena & Gopalani, Domain Independent Framework for Automatic Text Summarization. , 2015) found that the sentence location, the named entities, and the proper nouns are the most effective features in identifying the salient elements in the text.

Learning the best combination of features seems to be impossible because we need to test the possible features in all combinations and conditions. However, some of the researchers tried to investigate the most lucrative features. For example, Lin used the SUMMARSIT system, to generate summaries for multilingual input texts and to learn good combination functions (Lin C.-Y. , 1999).

The sentence centrality or sentence importance feature: this feature was firstly proposed by Yeh et al. ((Yeh, Hao-RenKe, Yanga, & Meng, 2005), and also used in (AbdelFattah & Ren, 2009), and (Ferreira, et al., 2013)). The sentence centrality measures the similarity between the sentence and the other sentences of the documents. In ((Yeh, Hao-RenKe, Yanga, & Meng, 2005), (AbdelFattah & Ren, 2009), and (Ferreira, et al., 2013)) the authors used simple vocabularies overlaps (Jaccard coefficient model) or simple statistical calculations to determine the sentences similarity. In this research, we found that the use of vocabulary overlaps hurt the condensation rate. Therefore, we used more accurate statistical approaches to measure the sentence centrality using the Vector Space Model and Latent Semantic Analysis.

Information retrieval techniques based on the VSM model: El-Haj and Hammo (El-Haj & Hammo, 2008) utilized the IR techniques to generate automatic summary. They employed the cosine similarity and weighted the document's terms based on the tf.idf scheme. El-Haj and Hammo produced focused and informative summaries based on the user query. They measured the cosine similarity between the user query and each sentence in the retrieved document. Ghwanmeh et al. (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced Algorithm for Extracting the Root of Arabic Words, 2009) used the first sentence as the base of similarity comparison. Ghwanmeh et al. computed the cosine similarity between the first sentence and each sentence in the documents. Ghwanmeh et al. assumed that the first sentence is the main sentence, but it usually is the introductory or the hook sentence, especially in news and media, that introduces the topic sentence or attracts the user's attention[8].

Information retrieval techniques based on bushy path and aggregate similarity: Bushy path and aggregate similarity are IR related terms investigated by Ferreira et al. (Ferreira, et al., 2013). the Bushy path method represented the sentence as a node on a map, two nodes (sentences) in the graph are connected if the similarity between them is greater than zero, and the similarity value between two nodes is placed above each link. The bushy path method

[8] See http://laflemm.com/reso/introSentences.html (Last update of this page: Feb. 27, 2014)

counts the number of the connections generated from each node and the node with the greatest number of connections takes the highest score. Aggregate similarity used the same graph idea, but it sums up the similarity values placed on the links. The disadvantage of the bushy path and aggregate similarity is related to the way they deal with links' labels; they maximize the labels of small similarities and equalize them with large similarities labels. For example, the bushy path gives more value to the node that has two links than the node that has one link even if the similarity in the latter case reaches 100%, and aggregate similarities give the node that has two links with aggregate similarities 70% more value than the node that has one link with 99% similarity.

Neural Network (NN): several techniques of NN have been experienced in the field of automatic text summarization. These methods techniques: the deep auto-encoder NN, the sequence to sequence, the recurrent NN, the Feed Forward NN and Gaussian mixture network, and the probabilistic neural network (Abdel Fattah & Ren, 2008), (AbdelFattah & Ren, 2009) , (Yousefi & Hamey, 2017), (Song, Huang, & Ruan, 2018).

Latent Semantic Analysis text summarization: The LSA is an advanced statistical model for analyzing the text contents semantically. It acquires knowledge and meaning through the correlation of facts from massive datasets (Ngoc & Tran, 2018), (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Wang & Ma, 2013), (Froud, Lachkar, & Ouatik, 2013), (Babar & Patil, 2015), (Ngoc & Tran, 2018). The LSA estimates the semantic meaning of the sentence through the exploration semantic context of the text. The LSA assumes that the text meaning is the collection of the meanings of the words found in the text and those meanings should be captured by analyzing huge text corpus (Yang, Bu, & Xia, 2012). It important to mention in this context that the researchers proved that the LSA is an effective tool in text extraction because it addresses the semantic meaning which solves some of the problems of the other statistical approaches (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Yang, Bu, & Xia, 2012), (Wang & Ma, 2013), (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015), and (Babar & Patil, 2015). Yates and Neto in (Yates & Neto, 1999) mentioned that the LSA was developed in the fields of IR and NLP to solve two main problems in the VSM model:

(1) Synonyms problem*:* synonyms arise when two or more words share a single meaning. For example, the Arabic words شُجاع, جَسُور, جَريء, and مقدام have single meaning "brave."

(2) Polysemy problem*:* Polysemy arises when a single word has several meanings. For example, the Arabic Word عين means allocated, eye, spy, appoint, and spring of water.

The employment of latent semantic analysis model appeared in the following research efforts:

- Mashechkin et al. (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011) used the LSA to generate generic extracts, and they integrated the LSA with non-negative matrix factorization to preserve the internal structure of the text.

- Yang, Bu, and Xia (Yang, Bu, & Xia, 2012) used LSA to reduce the effect of synonyms and polysemy problems generated from the use of VSM.

- Wang and Ma (Wang & Ma, 2013) added more semantic information to obtain accurate sentence selection when they chose the sentences that best describe the concept and contains certain terms that best represent it.

- Ba-Alwi, Gaphari, and Al-Duqaimi (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015) experimented the LSA for Arabic language, and they achieved 46% average ROUGE.

- Babar and Patil in (Babar & Patil, 2015) compared the LSA with Fuzzy logic in scoring and selecting the summary sentences and they found that the accuracy of the LSA was the highest.

- Ngoc and Tran in (Ngoc & Tran, 2018) integrate the LSA with Dennis coefficient to semantically classify the English text.

The LSA computes the similarity between two texts by identifying the shared concept. Therefore, it goes beyond the actual existence of the words and collects the words that have a single meaning in one semantic space. The LSA reduces the original terms-documents matrix to three matrices; the terms-concepts matrix, documents-concepts matrix, and the third matrix that represents the strength of each concept relative to each term and document. It uses the SVD to map the original terms-documents matrix to the above-mentioned matrices. The SVD is an algebraic matrix factorization technique that decomposes a rectangular, huge, and sparse matrix and produces a smaller matrix with a low rank. (Yates & Neto, 1999).

Table 2.8 Extraction Techniques with Precision

Ref	Technique	Features	Accuracy (R, P, f-score , CR)
(Luhn, The Automatic Creation of Literature Abstracts, 1958)	Statistical	Term frequency	Not mentioned
(Baxendale, 1958)	Statistical	Word position	Not mentioned
(Edmundson, 1969)	Statistical	Cueword/ Sentence location /Title&heading words	Not mentioned
(Lin C.-Y. , 1999)	Statistical	tf / tf.idf / Title and position / IR signature /Average lexical connectivity /Numerical data /Proper name, pronoun and adjective /Weekday and month	The maximum F_ Measure value is 58% at CR 20%
(Yeh, Hao-RenKe, Yanga, & Meng, 2005)	Statistical (MCBA + GA) (LSA + T.R.M)	Word position / Positive keyword / Negative keyword / Centrality / Resemblance to the title	At CR 30% f-score= 52% for CBA+GA, f-score =40% for LSA+TRM
(Yanmin, Bingquan, & Xiaolong, 2007)	Linguistic analysis	Locating the lexical chains obtained from HowNet and TongCiCiLin lexical databases.	At CR=10%, P=73%, R =77% At CR=20%,P=71%, R=74%
(El-Haj & Hammo, 2008)	Statistical Method	Term frequency / inverse term frequency	Not mentioned
(Svore, Vanderwende, & Burges, 2008)	Statistical with Machine learning	Position / N-grams frequencies / query term / Wikipedia entity(titles of Wikipedia pages)	CR = three sentence ROUGE-1 score = 52%
(Abdel Fattah & Ren, 2008)	Statistical with Neural Network	Sentence Position / Keywords /negative keywords / Centrality /Similarity to the title /Proper noun / Numerical data /Sentence length /Pushy path /aggregate similarity.	At CR=10%, R=45% At CR=20%, R=46% At CR=30%., R=47%
(AbdelFattah & Ren, 2009)	Statistical	position / positive keyword / negative keyword / centrality / sentence resemblance to the title / aggregated similarity / Inclusion of name entity, sentence / inclusion of numerical data / sentence relative length / Bushy path of the sentence.	Using DUC 2001 dataset (*The maximum precision was obtained using GMM)* P(GMM) = 60% CR=10% P(GMM) = 60% CR=20% P(GMM) = 60% CR=30%
(Shams, Hashem, Hossain, Akter, & Gope, 2010)	Merged statistical and linguistic methods	Statistical parameters: tf / Sentence weight / Subject weight. Linguistic methods: Employed Stanford POS Tagger and a term co-occurrence graph in order to find the subject of the sentence.	At CR= 30% R=65%
(El-Shishtawy & El-Ghannam, 2012)	Statistical and Linguistic	Normalized Phrase Words / Phrase Words / Phrase Relative Frequency / Word Relative Frequency. / Sentence Location /Phrase Location /Phrase Length /Contain Verb /Is It Question	At CR=25% R=52% P=71%
(Azmia & Al-Thanyyan, 2012)	Statistical- Rhetorical Structure Theory	word frequency / sentence location / title keyword	At CR=31% P=66% R=70% F- Measure =67%
(Alruily, Hammami, & Goudjil, 2013)	linguistic methods	Transitive verbs by prepositions.	Not mentioned
(Ferreira, et al., 2013)	Statistical	tf.idf / Upper case /Proper noun / Word co-occurrence / Lexical similarity Cue-phrase /Inclusion of numerical data / Sentence position /Sentence centrality / Resemblance to the title / Aggregate similarity/Bushy path	AR = 73% AP = 40% AF= (73%)
(Kiyoumarsi, 2015)	Statistical- Machine learning using Fuzzy and Vector methods	Mean-tf-ISF / Sentence-to-Sentence cohesion / Sentence to centroid cohesion	Vector Method: At CR=10% R=21%, P=21% Fuzzy Method: At CR=10% R=28.2% P=29.6%
(Babar & Patil, 2015)	Statistical Methods with Fuzzy logic and LSA.	Title words / Sentence position / Sentence length / Numerical data / Thematic words / Sentence to sentence similarity / Term weight / Proper nouns	CR not computed Using fuzzy scoring: R=41%,P=86% Using LSA:R=44%,P=90%
(Chen, et al., 2015)	Statistical, with the recurrent NN	Tem frequency in the sentence / The sentence length	CR not computed ROUGE 0.1 R=40%, ROUGE 2.0 R=26%
(Yousefi & Hamey, 2017)	Statistical, with the NN	Term frequency	average ROUGE 46%
(Tayal, Raghuwanshi, & Malik, 2017)	Linguistic and Statistical	Word tag(Subject, Verb, and Object) / Title or theme of the document / N-gram co-occurrence	CR Not mentioned f-score = 14%, R= 40%
(Al-Radaideh & Bataineh, 2018)	Statistical with genetic algorithm	Term frequency, Sentence position, Sentences length, similarity to the title	At CR=40%, Avg R=55%, AP=45%, f-score =54%.

The SVD is a powerful and effective reduction, but it has a tangible drawback related to the huge space and time complexity requirements. Donga et al. (Donga, Haidar b, Tomov b, & Dongarra, 2018) showed that 70%-90% of the execution time of the modern applications that use the LSA goes to the running of the SVD. He et al. (He, Deng, & Xu, 2006) detailed the time complexity analysis of the LSA Similarity; they found that the time complexity is the minimum of $\{t^2d, td^2\}$ where t is the number of terms in huge corpus and d is the number of documents.

2.3.2.2 Linguistic Approaches

The Linguistic approaches are language-dependent, and they extract the summary based on the linguistic features or structures (see [(Yanmin, Bingquan, & Xiaolong, 2007), (El-Shishtawy & El-Ghannam, 2012), and (Alruily, Hammami, & Goudjil, 2013)]).

The vast majority of the research reported earlier relies on statistical approaches, Yanmin, Bingquan, and Xiaolong (Yanmin, Bingquan, & Xiaolong, 2007) followed another direction toward the linguistic analysis. They investigated the cohesion structure of the text by locating the lexical chains obtained from HowNet and TongCiCiLin lexical databases. Tayal et al. (Tayal, Raghuwanshi, & Malik, 2017) used the POS tagger and NLP parser to analyze the sentence before finding its semantic meaning using WordNet. Alruily, Hammami, and Goudjil (Alruily, Hammami, & Goudjil, 2013) utilized a linguistic feature of the Arabic language to delete all the text located between the verb and its object which takes the form of a preposition phrase. Shams, Hashem, Hossain, Akter, and Gope (Shams, Hashem, Hossain, Akter, & Gope, 2010) merged statistical and linguistic methods in one summarization system. They employed Stanford POS Tagger and a term co-occurrence graph to find the subject of the sentence. However, the linguistic approaches are language-dependent, and almost we cannot generalize and use them for another language.

2.3.3 ATS evaluation

The assessment of the extract relevance quality is an important issue related to the ATS (or even the text mining in NLP). As stated by Jing et al. (Jing, Barzilay, McKeown, & Elhadad, 1998), we cannot assume the existence of a typical answer and use it to evaluate the results. Sparck and Galliers (Sparck & Galliers, 1995) and Mani (Mani, Automatic Summarization., 2001) stated that the evaluation strategy for any summarization systems should include means to measure:

- The summary length or the condensation rate: This equals the summary length divided by the full-text length.

- The salient parts: these measures if the automatic summary preserves and maintains the main ideas found in the original text.

As mentioned by Sparck and Galliers (Sparck & Galliers, 1995), the two main approaches experienced to evaluate the quality of automatic summaries are Intrinsic and Extrinsic approach.

- The intrinsic approach uses the human-generated summary as an ideal answer and compares the system generated summary against the human-generated summary to find the resemblance between them. Recall and precision are the primary measures of the intrinsic approach. The intrinsic approaches are used in the following references [(Edmundson, 1969), (Kupiec, Pedersen, & Chen, 1995), (Yanmin, Bingquan, & Xiaolong, 2007), (Mihalcea & Ceylan, 2007), (El-Haj & Hammo, 2008), (Binwahlan, Salim, & Suanmali, 2009) , (Al-Radaideh & Bataineh, 2018)].

- The extrinsic approach assesses the impact of the automatic summary on the other NLP fields such as Topic Detection, Question Answering systems, and Information Retrieval [(Chen, Wang, Liu, & Wang, 2002), (Harwath & Hazen, 2012)].

2.3.3.1 ATS Automatic Evaluation Tools

The intrinsic manual approach is widely used in the past, but this type of evaluation is affected by many factors such as the evaluators' backgrounds and education levels (El-Haj & Hammo, 2008) and the evaluators' opinions (or point of view (Halteren & Teufel, 2003)), In addition, it is expensive and takes a lot of time. Therefore, the researchers implemented the intrinsic approach in many evaluation tools.

- ROUGE (Lin C. Y., 2004): ROUGE Evolution Toolkit is an evaluation software developed by Lin (Lin C. Y., 2004). It stands for Recall-Oriented Understudy for Gisting Evaluation, and it evaluates the quality of automatically generated summaries by comparing them with human-generated summaries (called reference, gold, or Ideal summary). ROUGE counts the number of intersections between the computer-generated summary and the gold summary created by

humans (or by another system for comparison purposes). ROUGE statistically measures the resemblance, but it cannot indicate the percent of complete sentences from the gold summary appear in the automatic summary. The produced recall and precision measurement values increase by the existence of any sequences of n-grams, words, or phrases.

- Summary Evaluation Environment (SEE): SEE was developed by Lin (Lin C. , 2001), and it stands for Summary Evaluation Environment. It is a Software package that facilitates the evaluation of computer-generated summaries. It provides an interface with two panels, one shows the computer-generated summary (called peer summary) and the second shows the reference summary (called model summary). Assessors evaluate each sentence in the peer summary panel and then compare it with the model summary. Each sentence in the peer summary takes one of five values (All, Most, Some, Hardly, and None) depends on the degree of similarity to the model summary sentences. The assessors evaluate the summary contents, grammar, coherence, and cohesion. The tool facilitates the manual evaluation.

- MeadLeval: The MeadLeval toolkit employs a data structure called the extract file; this file stores important information about the sentences contained in the extract. Similar to ROUGE and SEE, MeadLeval compares the computer-generated summary with the ideal summary. MeadLeval supports many evaluation metrics recall, precision, Kappa, and others.

2.4 Automatic Synonyms Extraction

The idea of constructing a lexical database came from a group of psychologists and linguists who aimed to find an informative way to search English dictionaries (Miller, Beckwith, Fel, Gross, & Miller, 1990). They manually collected the synonyms and stored them in a lexical database. This database grouped words based on their meanings (the synonym relation) and the grouped words called synsets. The synsets linked together via Super-subordinate relation in which the general objects belonging to certain synset linked to more specific object belongs to anther synset. Also, the relation was transitive, which allowed the relation to link general synsets with the parts of the specific synsets (Fellbaum, WordNet and wordnets, 2005).

2.4.1 Synonyms Sets Creations and their Influence

Researchers in Computational Linguistic interested in investigating the synonyms of the words and they arranged them in a special kind of dictionaries called WordNet (Fellbaum & Vossen, Connecting the Universal to the Specific: Towards the Global Grid, 2007). The WordNet is an extensive database storing the words together with their synonyms. It is a concept dictionary that groups words based on their meanings to produce synonyms sets. The first appearance of WordNets was at Princeton University, and it performed manually for the English language (Miller, Beckwith, Fel, Gross, & Miller, 1990). According to Miller et al. in (Miller, Beckwith, Fel, Gross, & Miller, 1990), the WordNet aimed to facilitate the searching in the dictionary and to substitute the regular word searching -which is typically done by the alphabetical ordering of words- by the concepts searching.

The EuroWordNet was developed for eight European languages. The EuroWordNet used the same model used to construct the Princeton WordNet, and it added two new contributions, the hiring of the Base Concept and the addition of new relations with a precise way to clarify the relations among the synsets (Fellbaum & Vossen, Challenges for a multilingual wordnet, 2012). AWN is the WordNet developed for the Arabic Language (Elkateb, et al., 2006). Elkateb benefited from the model used in Princeton WordNet and EuroWordNet, but they faced real challenges related to the morphological structure of the Arabic Language. To solve these challenges, Elkateb combined the Interlingual Index used in EuroWordNet with the suggested upper merged ontology.

In many NLP publications, the semantic investigation of the text contents was improved by hiring a semantic dictionary such as the synonyms dictionary in the investigation process. The term weight of a given term is computed based on its parameters (for example, term frequency and inverse term frequency) and the parameters that can be obtained from its synonyms. In the field of text classification, Scott and Matwin in (Scott & Matwin, 1998), used the WordNet and computed the weight of a term by dividing the number of occurrences of the term synsets (taken from WordNet) in the document over the document length. Bloehdorn and Hotho in (Bloehdorn & Hotho, Boosting for text classification with semantic features, 2004) used WordNet to generalize the terms to their concepts and employed them in the classification process instead of individual terms. In (Bloehdorn, Basili, Cammisa, & Moschitti, 2006), the authors mapped the terms to their super concept using WordNet. Another semantic text repository used to improve the semantic text classification was Wikipedia (Wang & Domeniconi, 2008) and Open Direct Project (Evgeniy & Shaul, 2007).

Text Categorization is another field of text mining utilized WordNet semantic dictionary, in (Jianqiang, Yu, & Bo, 2009) the authors used WordNet to build training data and in (Barak, Dagan, & Shnarch, 2009) the authors supplemented the Latent Semantic Analysis with concepts extracted from WordNet. In (Luo, Chen, & Xiong, 2011), Luo et al. proposed a weighting scheme that multiplies the tf of a term by semantic similarity value of that term with the terms found in the name of the category and its interpretations that are taken from the WordNet. The major benefit they gained was the smaller training data required to distribute the uncategorized document over their categories.

In the field of Information retrieval, the semantic investigations of the term synonyms and the generalization of terms to concepts during the weighting and indexing process addressed by many researchers. In (Dinh & Tamine, 2015), Dinh and Tamine used semantic meaning to solve the ambiguities that are generated from the regular tf.idf weighting by mapping the terms to specific concepts taken from MeSH (semantic dictionary for medical data) and then correlated the concept to the correct domain. In this case, the IR system can capture the correct meaning of the term because it knows the concept underlying the term and to which domain the term belongs. Dinh and Tamine built an information retrieval system and implemented their idea, and they gained a noticeable improvement in the relevancy of their system comparing with two baseline information retrieval systems. However, the authors used specific domain knowledge, which is the biomedical documents, that enabled them to reduce the problem (a small number of concepts and a few numbers of domains).

2.4.2 Synonyms Extraction Techniques

In the literature of the ASE, three main extraction techniques can be derived from the published research:

1. The statistical techniques over monolingual corpora, such as the Vector Space Model with cosine similarity or relative cosine similarity (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), (Crouch, 1990), (Chen & Lynch, 1992).
2. The translation techniques among different languages over the bilingual or multilingual dictionaries (the words that share the same interpretations are synonyms) (Lin, Zhao, Qin, & Zhou, 2003), (Lonneke & Jorg, 2006), and (Ageishi & Miura, 2010).
3. The linguistic analysis techniques that syntactically and semantically parse the corpus or the dictionary to extract synonyms (Minkov & Cohen, Graph based similarity measures for synonym

extraction from parsed text, 2012), (Henriksson, Moen, Skeppstedt, Daudaravicius, & Duneld, 2014), (Benabdallah, Abderrahim, & Abderrahim, 2017), (Grefenstette, 1994), and (Senellart & Blondel, 2004).

2.4.2.1 Cosine Similarity-Based Synonyms Extraction

Many research publications employed the VSM model that is used in the field of information retrieval in the synonyms extraction method. The tf.idf weighting scheme and the cosine similarity are adapted to reflect the terms to terms relations instead of the query to document relations in the IR. Analog to VSM model in IR, the CBoW and SG model are developed in the field of synonyms extraction, and those models hired the cosine similarity to find the similarity between specific term and all the terms found the corpus (Mikolov, Chen, Corrado, & Dean, 2013).

Chen and Lynch (Chen & Lynch, 1992) used the Vector Space Model to extract the synonyms by computing the cosine similarity between the terms found in a document within a large corpus. They collected the nouns (called the descriptors) and computed the cosine similarity between all the descriptors that have a frequency greater than 3. Before Chen and Lynch, Crouch (Crouch, 1990) built an automatic thesaurus dictionary to expand the user query in information retrieval research. Crouch used the VSM and represented the terms as dimensions, and the documents were vectors in the term dimension space. The author built thesaurus classes by combining similar documents in one cluster. Then, Crouch extracted the terms that had a low document frequency from each cluster to form the thesaurus classes. The aim was not the synonyms by themselves, but to expand the user query terms with supporting terms found in the similar documents.

Leeuwenberga et al. in (Leeuwenberga, Vela, Dehdar, & Genabith, 2016) emphasized the idea that the simple cosine similarity hurt the precision because it combines synonyms, hypernym, and hyponyms in the synonym set. Leeuwenberga et al. proposed to consider the top ten similar words and included them in the calculation of the similarity to obtain more accurate similarities. They divided the simple cosine similarity between w and w_q words by the summation of the cosine similarities of the top ten words similar to w. Leeuwenberga et al. obtained 12% precision value (for both English and German). To solve the problem of computation penalty, Zhang and Wang in (Zhang, Li, & Wang, 2017) used the Word2Vec model based on CBoW and SG model to map the relations among

the corpus terms, and used the cosine similarity to find the similarity, and used spectral clustering to identify the synonyms.

The authors of (Henriksson, Moen, Skeppstedt, Daudaravicius, & Duneld, 2014) developed a model for extracting synonyms in biomedical data, and they aimed to enhance the distributional hypothesis model. The distributional hypothesis is a semantic model developed by Harris in 1954 (Harris, 1954) and stated that synonyms have a convergent language distribution and share the same context. The distributional hypothesis model deeply used to extract the related meaning words, but Henriksson et al. enlarged the scale of the semantic relations among the terms by incorporating two distributional models-instead of one as usual in the distributional hypothesis – and two corpora instead of one large corpus. The aggregation of multiple models and corpora enabled the authors to create more semantic spaces which enriched the relations between the terms.

Recently, and to build an ontology for Arabic Language, Benabdallah et al. depended on stored patterns (called them markers) to find the semantic relations between statistically selected terms (Benabdallah, Abderrahim, & Abderrahim, 2017). AlMaayah et al. In (AlMaayah, Sawalha, & Abushariah, 2016) produced a synonym set for the terms of AlKuran AL Kareem (the holy book for Muslims). AlMaayah produced the synsets by linking the Quran's terms with their meanings obtained from a traditional dictionary. The authors succeeded to improve the recall of the semantic search by around 27% compared with a baseline system. Table 2.9 shows the list of references that use the statistical methods with their models and some accuracy results collected from them.

Table 2.9 Summary of the Statistical Models with their Accuracy (Found in Related Work)

Ref	Statistical Model	R	P
(Chen & Lynch, 1992)	Cosine similarly (CBoW and SG model) Cluster Algorithm	27% 35%	62% 66%
(Leeuwenberga, Vela, Dehdar, & Genabith, 2016)	Relative cosine similarity model	7% German 12% English	12% German 12% English
(Zhang, Li, & Wang, 2017)	Word2Vec model/ CBoW and SG model/ Cosine Similarity/ Spectral Clustering	74% Manual	80% Manual
(Henriksson, Moen, Skeppstedt, Daudaravicius, & Duneld, 2014)	Enhanced distributional hypothesis model	47%	8%
(Benabdallah, Abderrahim, & Abderrahim, 2017)	Learning Extraction Markers tf.idf weights	84% Manual judgment, domain-specific corpus	76%
(Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012)	Path Constrained Graph model	MAP = 59%	
(Lonneke & Jorg, 2006)	Distributional Similarity model	13%	23%
(AlMaayah, Sawalha, & Abushariah, 2016)	tf.idf weights (VSM)	35%	33%

2.4.2.2 Syntactical Analysis Based Synonyms Extraction

The linguistic analysis that syntactically and semantically parses the corpus or the dictionary to extract synonyms are also experienced in the field of synonyms extraction. Grefenstette in (Grefenstette, 1994) made a syntactical analysis including tokenization, proper noun detection, part of speech tagging, part of speech disambiguation, and parsing. Grefenstette extracted the noun and verbs phrases and parsed them to extract the syntactic relations among the terms composing those phrases. Then, for any two nouns, Grefenstette computed the similarity by tracing the common modifiers between them (between the two nouns being processed). The modifiers could be nouns, verbs, or adjectives.

Another syntactic approach proposed by Lobanova et al. in (Lobanova, Spenader, Cruys, Kleij, & Sang, 2009), their idea was to improve the precision through the elimination of antonyms that might appear in the synonym list. They used two techniques to investigate the semantic relations between the terms and find the antonyms, the first one used two manually selected patterns, and the other used Ravichandran and Hovy method to learn the antonyms through the scanning of the corpus automatically and identifying the lexical relations between the pairs of words.

Senellart and Blondel in (Senellart & Blondel, 2004) used a graph-based approach to find similar words. They created a graph for every word *w* found in Webster's dictionary. The graph links *w* with every word appeared in *w's* definition and links it with every word the word *w* appeared in its definition. After that, a subgraph is created for the query word, and the similarity between this subgraph and the dictionary graph *i* is determined, and the parts of the dictionary graph that resemble the subgraph is extracted. Senellart and Blondel restructured the base dictionary in a way that allows them to discover the relations among the dictionary words. The use of the graph-based extraction of synonyms also hired by Minkov and Cohen in (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012); the similarity was obtained by restructuring the parsed text in a walk graph. The walk graph used to extract named entity, and Minkov and Cohen showed it could present good results in the field of synonym extraction.

Opposite to the statistical method, the syntactical methods are language-dependent because they depend on the grammatical and lexical rules of a specific language. Also, it requires the use of a language-specific dictionary, Parser, POS tagger, and Tokenizer.

2.4.2.3 Translation based synonyms extraction

Other research works used automatic translations to extract synonyms. Bilingual or multilingual dictionaries or statistical machine translation are used to do the translation between the languages and the words that share the same interpretations or translation are considered as synonyms. Lin et al. in (Lin, Zhao, Qin, & Zhou, 2003) measured the similarity among the translations generated from bilingual dictionaries to extract the semantically related words, whereas, in (Lonneke & Jorg, 2006) the authors used multilingual corpus (containing 11 languages) to find the words that share the same translation context. In (Ageishi & Miura, 2010), Ageishi and Miura used statistical machine translation to obtain domain-specific synonyms. The translation probabilities were computed between the pairs of sentences taken from two datasets, and the terms with high probability value are considered as synonyms.

2.4.3 Main Findings

The initiating of this research requires holding a deep reading about what has been achieved in the field of Arabic information retrieval. In this chapter, a quantitative relevancy survey to measure the enhancements achieved has been established. The survey reviewed the impact of statistical and morphological analysis of Arabic text on improving the Arabic IR relevancy. The survey measured the contributions of Stemming, Indexing, Query Expansion, Automatic Text Summarization, Text Translation, and Named Entity Recognition in enhancing the relevancy of Arabic IR. Our survey emphasized the quantitative relevancy measurements provided in the surveyed publications. The survey showed that the researchers achieved significant enhancements, especially in building accurate stemmers, with accuracy reaches 97%, and in measuring the impact of different indexing strategies. Query expansion and Text Translation showed a positive relevancy effect. However, other tasks such as NER and ATS still need more research to realize their impact on Arabic IR.

Regarding the automatic text summarization using LSA model, we found that in (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Yang, Bu, & Xia, 2012), (Wang & Ma, 2013), (Froud, Lachkar, & Ouatik, 2013), (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015) , (Babar & Patil, 2015), (Ngoc & Tran, 2018), the time complexity of running the LSA procedure was not addressed which represents the main challenge of employing the Latent Semantic Analysis in any NLP application. Recently in 2017, Gambhir and Gupta (Gambhir & Gupta, 2017) reviewed almost all the automatic text extraction techniques proposed in the literature. They listed the published papers with their approaches and results.

The survey showed that great effort spent, but the research that addressed the semantic analysis did not take into consideration the high time complexity required to apply the LSA on text extraction. Therefore, the surveyed publications represent a good starting point because they proved that the LSA is a powerful mechanism to extract a summary from the text.

Regarding the automatic synonyms extraction, the main aim of the synonyms extraction method proposed in this introductory chapter is to give more accurate synonyms in a reasonable time interval. All the mentioned publications in the literature review chapter that investigated the synonyms extraction chapter either require heavy access to a stored database of terms (monolingual, bilingual, or multilingual dictionaries), relations' patterns or require massive computational operations over all the terms found in a huge corpus. In our work, we do not use the idea of a base dictionary to weight or extract the terms, and we extracted the semantic relations between the nouns by only considering the verbs. Also, not all verbs will be processed, the verbs that have a large term frequency or appeared with a large number of nouns will be neglected.

In the next chapter, we continue from what we found in the literature review, and we will build an extraction system (MLSExtractor) that uses the LSA in an efficient way. The MLS is a reduction step applied to the original terms-sentences matrix and produces a lower-dimensional matrix. The MLS precedes the SVD execution, so its effect is reflected directly on the execution time of the SVD. Also, the next chapter describes the method used to efficiently extract the term synonyms and how we boost the user query with a list of synonyms for each word mentioned in that query.

CHAPTER 3 METHODOLOGY

This chapter explains the method used to satisfy the main aim specified in chapter 1. The aim is to improve information retrieval efficiency and performance. The goal of our method is to design a solution of the large size inverted index in the information retrieval applications and to boost the user query terms with semantically related terms. The method used efficient statistical and semantic models to reduce the size of the original inverted index to a short and informative inverted index.

The method includes three phases, phase one includes the design of the automatic extraction system that extracts the salient parts of the documents before the indexing process is initiated, phase two describes the design of the automatic synonyms extraction system to extract the query semantically related words, and phase three describes the design of an information retrieval system based on the VSM model.

3.1 Introduction

The VSM model is the most commonly used in the IR field, and this is why we choose this model. We choose the model that already used and tested for a long time. Note that our target is to test the effect of semantic text extraction in reducing the inverted index and how this reduction affected the IR relevancy and space efficiency. Therefore we used a well-known and tested IR model.

3.2 General Architecture

Figure 3.1 gives a general overview of the method developed in the research. The input of the method is a huge number of text documents named D_1, D_2, D_3 ... , D_n. These documents comprise 40,006 documents that were taken from Essex, Kalimat, 242, and Blog Authorship datasets, as described in section 4.2.1. The output is a set of retrieved document RD_1, RD_2, RD_3,..., Rdi that are retrieved based on a VSM matching model. In the beginning, we want to mention that the red lines in Figure 3.1 follow the steps proposed in this research, and the green lines follow the traditional steps in the IR system. The green lines are inserted to the figure just to mention that a comparison will be established between our method of retrieval and the traditional retrieval in the VSM model. The three phases that are mentioned in the introduction of this chapter are represented in three models in Figure 3.1, the MLS extractor, the NBDV synonyms extractor, and the IR system.

The MLS extractor: It is a model for extracting generic summaries from the text documents by deleting the repetitive sentences in the document. The repetitive sentences are determined by measuring the verbatim, statistical, and semantic resemblance between any two sentences or paragraphs. We consider the MLS as a self-extraction system that extracts the main sentences without the influence of the linguistic features, text structure, and user intervention. Therefore, it is a language, domain, and user-independent extraction system. The MLS stands for Multi-Layer Similarly and the abbreviation summaries the MLS similarity computation strategy. The MLS computed the similarity between two pieces of text by using three statistical techniques, the Jaccard coefficient to process the verbatim similarity (lowest layer), the VSM to process the cosine similarity (middle layer), and the LSA to process the semantic similarity(upper layer). The detailed method of the MLS extraction explained in section 3.3.

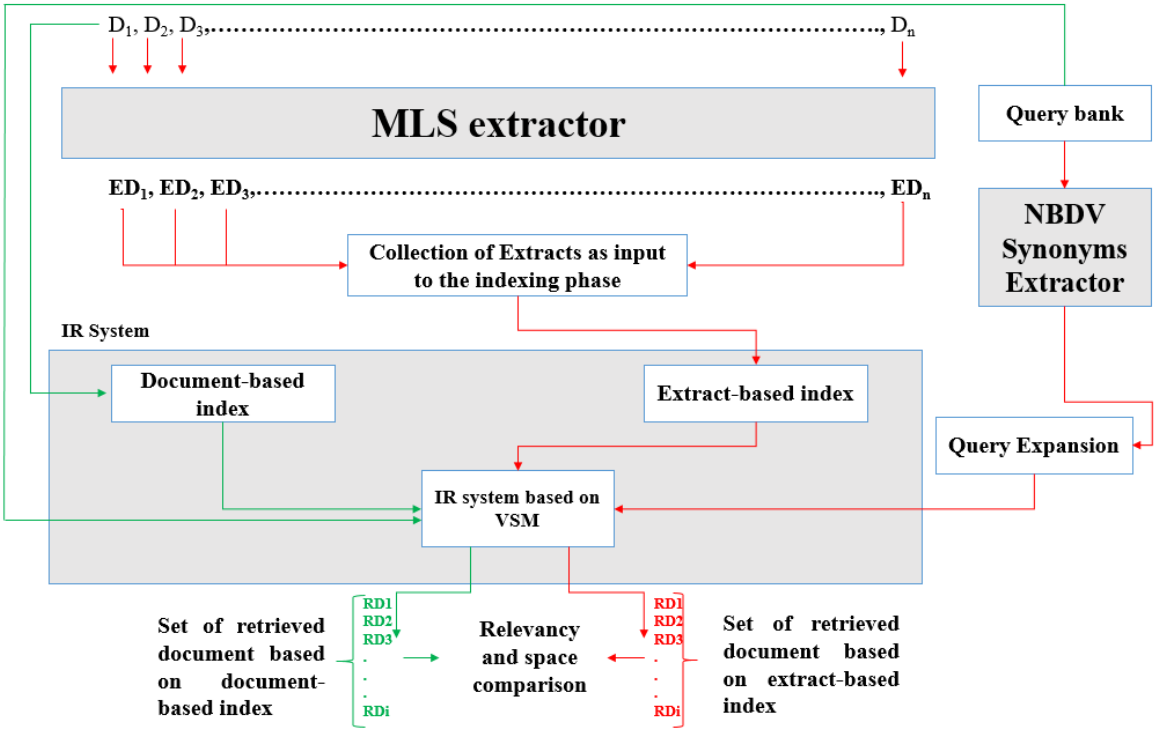


Figure 3.1 the IR System with MLS Extractor and NBDV Synonyms Extractor

According to the categorization of the summaries stated in (Mani, Automatic Summarization., 2001), (Mei & Chen, 2012), and (Gambhir & Gupta, 2017), we classify the automatic generated summaries by the MLS model as:

(1) Extracts: because we copy certain parts from the original document, and we do not restructure the sentence or change their order.

(2) Informative: because we try to extract all the salient parts of the text, not parts of them.

(3) Free size text: because we work on the sentence level, and any text contains two sentences or more can be processed by our method of extraction.

(4) Generic: because the extraction process does not focus on certain factors such as the user query, the document title, or the key terms.

The NBDV Synonyms extractor: it is a model that extracts the synonyms or the related meaning words of a noun based on the semantic investigation of the relations between the nouns. The NBDV uses an efficient weighting scheme called the Orbit Weighting Scheme to weight the distinctive verbs shared between groups of nouns. The OWS is proposed to handle the time efficiency problem of the traditional tf.idf weighting scheme. The NBDV extractor is used to boost the user query terms with semantically related terms. The detailed method of NBDV extraction explained in section 3.4.

The IR system: the IR system designed in this method is a traditional IR system based on the VSM model. We choose a traditional IR system because we are not developing the matching strategies between the document terms and the query terms, we are solving the large size problem of the inverted index by semantic summarizer that summarizes the original documents. Any loss of information caused by our summarizer is rectified by boosting the query terms with semantically related words. The detailed method of the IR system explained in section 3.5.

Figure 3.1 shows two inverted indexes, the extract-based inverted index, and the document-based inverted index. The extract-based inverted index is the inverted index generated after summarising the original documents by the MLS extractor, and the document-based inverted index is the inverted index of the original documents without summarization. Thus, the architecture shown in Figure 3.1 depicts the IR process with and without text summarization.

3.3 Automatic Text Extraction Method

This section describes our method of the Automatic Text Extraction that produces generic summaries of a text document. The method statistically measures the verbatim, statistical, and semantic resemblance between any two sentences or paragraphs and deletes the repetitive sentences. The method is designed to work with a single document or multi documents because it can measure the similarity at the sentence level (or any two segments of text).

In the design of our model of text extraction, we do not use reference sentence as a base for the extraction (such as the user query (El-Haj & Hammo, 2008) , the first sentence (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced Algorithm for Extracting the Root of Arabic Words, 2009), or the title of the document (Edmundson, 1969), (Lin C.-Y. , 1999), (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (Abdel Fattah & Ren, 2008)). This feature of our method makes it more flexible in dealing with any kind of text (news, books, articles, or others) because the system will not oblige to take a certain direction during the extraction process. Thus, the output includes a variety of information depending on what the document contains[9].

Another important issue is the condensation rate or the summary length. The CR is normally fixed as in (Marcu, 1998), (Douzidia & Lapalme, 2004), (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (AbdelFattah & Ren, 2009), (El-Shishtawy & El-Ghannam, 2012), (Al-Radaideh & Bataineh, 2018), or user predetermined as in (Hassel, 2004), (Azmia & Al-Thanyyan, 2012). In our method, the condensation rate depends on the amount of similarity between the documents' sentences, thus the output is a variable-sized summary that contains the main ideas found in the documents. The fixed condensation rate forces the system to return a certain number of sentences or a predetermined ratio of the text and this may cause the systems to neglect certain salient sentences because the summary length exceeded the condensation rate limit. Our claim states that the condensation rate should depend on the richness of information found in the document and our algorithm implements this idea. During the explanation of our method of extraction, we will assume that the condensation rate is a feature of evaluation not a predetermined parameter.

3.3.1 Basic Concepts

The design of our method involves three salient parts, the weighting of the document's term, the similarity estimation between any two sentences in the text, and the deletion of repetitive sentences. In the beginning, Important definitions for computing the terms' weights and similarities are introduced. Then, the definition and lemmas necessary to select the extract's sentences are presented.

The method of extraction developed in this research estimates the similarity at four levels of complexity, the rate of sentence verbatim existence (Jaccard coefficient with no terms weighting), traditional statistical (vector space model

[9] Parts of this section and its subsections are mentioned in the second paper of the "**Publications Arising from This Thesis**" section

with cosine similarity and tf.idf terms weighting scheme), statistical with semantic analysis (latent semantic analysis with its classical definition), and multi-layer of statistical and semantic analysis (the multi-layer similarity model).

The first level of similarity estimation uses the Jaccard model: The Jaccard coefficient is used as the first level of similarity computation because it can process the parts of the text that contain a sufficient number of shared terms (Guha, Rastogi, & Shim, 2000). The Jaccard Coefficient has been used deeply to measure the text-similarity in many NLP and IR fields, and showed efficient and reasonable performance (Al-Kharashi & Martha, Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System, 1994), (Guha, Rastogi, & Shim, 2000), (Deng, Stefan, & Sergej, 2012), (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013). In Jaccard Similarity calculation, simple statistical calculations measure the percentage of shared terms between two sentences (Deng, Stefan, & Sergej, 2012), and because the number of words in the two sentences will not normally be large, the application of the Jaccard coefficient will be very efficient.

Definition 1. Given a text document T as a set of sentences,
$T = \{S_1, S_2, S_3, \dots, S_M\}$, and $S_i, S_j \in T$ are two sets of words (terms) such that $S_i = \{t_1, t_2, t_3, \dots, t_x\}$ and $S_j = \{t_1, t_2, t_3, \dots, t_y\}$, if $S_i \cap S_J \neq \emptyset$.

Then, the Jaccard similarity is defined by the following equation:

$$\mathrm{Jac}(Si, Sj) = \frac{|Si \cap Sj|}{|Si \cup Sj|} \qquad (1)$$

Example1: doc 9 in Essex corpus: Given

$S_2 = \{$ ↑ زوج, رحل, ↑ عصي, ↑ رحب, خوي, صور, عرف, خوي, رحب, ثور, ↑ علم, ↑ موسيقى, ↑ غني, ↑ عرب, ↑ شكل $\}$

and

$\boldsymbol{S_{11}} = \{$بعد, وفي, زوج, عصي, عام, خوض, جرب, عدد, جمع, لحن, ألف, برز, فلم,
وهب, برز, غني, كبر, جمع, قدم, رحب, زود, أبن, روس, شكل, عمل, صوف, زكي, وهب,
قدر, خلق, نمط, موسيقى, خوص, سقي, موسيقى, عرب, موسيقى, علم, وما, زول,
سور, فنن, سمر, نجح$\}$

$$\boldsymbol{Jac(S_2, S_{11}) = \frac{8}{51} = 0.15}$$

Note that the two sentences have different lengths so for length normalization we changed equation 1 to be:

$$Jac(S_i, S_j) = \frac{|Si \cap Sj|}{\min_{S_i, S_j \neq \emptyset}(|S_i|, |S_j|)} \quad (2)$$

Back to example1.

$$Jac(S_2, S_{11}) = \left(\frac{8}{15}\right) = 0.54$$

This equivalent to saying that 54% of the terms of S_2 are found in S_{11}.

The Jaccard similarity does not consider the words orders so the sentence "Ali beats Oqla" is completely similar to the sentence "Oqla beat Ali", thus the semantic meaning of the two sentences is not investigated. Also, the Jaccard coefficient does not consider the importance of the term with respect to another term or the whole document. The Jaccard sees the sentence as a bag of words without considering the terms meaning, orders, or relationships. However, we employed the Jaccard coefficient because the terms' overlaps, in some cases, can give a significant indication about the similarity if the overlap was large, and this can be happen for certain parts of the text.

The second level of similarity estimation uses the VSM model: The selection of the VSM model in the second level of similarity computation was based on three reasons:

1. In the literature of NLP and IR, the VSM text-similarity has been intensively used to estimate the similarity between text segments (text documents, text document and query, paragraphs, or sentences) (Salton, Wong, & Chungshu, A vector space model for automatic indexing, 1975), (Harrag, Aboubekeur, & Eyas, 2008), (Chen & Chiu, 2011), (Singh & Dwivedi, 2013), and (Mikolov, Chen, Corrado, & Dean, 2013).
2. The VSM showed superior precision over the Jaccard and Euclidean text-similarity models as concluded by Subhashini and Kumar in (Subhashini & Kumar, 2010).
3. In (Slamet, et al., 2018), the author used the VSM to obtain abstracts from scientific journals and the results were significant.

In any text mining field, the VSM requires two things; determining the weights of the terms and applying the cosine similarity. The VSM estimates the weight of the terms based on their frequency in the text's segments and their distribution over the whole segments found in the document(s). The parts of the text targeted by the second level of our analysis are the segments that contain terms that appeared frequently in those segments and distributed over a few numbers of segments (El-Haj & Hammo, 2008). In the VSM model, the terms that distribute in every segment is considered meaningless unit, such as the stopwords. The VSM claims that the word that represents a

topic or concept will not appear everywhere in the text. For example, the word "network" will appear mainly in the documents that talk about computer networking and this word will not be common in other fields.

In the employment of the VSM, the sentences in the document are represented as vectors in the vector space and the similarity between two vectors is determined by measuring the angle between them. Small-angle means high similarity. The VSM used the cosine to estimate the angle value (0<= cosine(angle) <=1), if the cosine was large (approaching to 1), this indicates to small angle and high similarity.

The contents of each vector are the weights of the sentence's terms. The term weight is computed based on term frequency (tf) and the document frequency (idf), which means the number of documents that contain the term. Certain weighting scheme should be hired, and we used the tf.idf weighting scheme that was proposed by Salton (Salton & McGill, Index construction, 1983).

Definition 2. For any sentence $S_i \in T$, the $tf_{t,si}$ is the number of times the term t appeared in the sentence S_i.The log of 10 normally normalizes the tf because the importance of the term does not increase proportionally with the tf. The most common formula used to compute the tf is:

$$\mathbf{tf_{t,Si}} = \begin{cases} \mathbf{1 + \log_{10} tf_{t,si}}\ , \mathbf{if\ tf_{t,si} > 0} \\ \mathbf{0} \qquad\qquad\qquad \mathbf{otherwise} \end{cases} \qquad (3)$$

Definition 3. Given a term t, the idf_t is the number of sentences in T that contain t. and the idf_t is given by the following equation:

$$\boldsymbol{idf_t = log\left(\frac{N}{df_t}\right)} \qquad (4)$$

Where N is the number of sentences in T

Definition 4. Given the $tf_{t,Si}$ and idf_t , the tf.idf weights $w_{t,si}$ of the term t is given by equation 5:

$$\boldsymbol{w_{t,si} = idf_t \,.\, tf_{t,Si} = log\left(\frac{N}{df_t}\right).\left(1 + log_{10}\, tf_{t,si}\ \right)} \qquad (5)$$

Definition 5. Given a text document T as a set of sentences and $T = \{S_1, S_2, S_3, \dots, S_M\}$, and given a sentence S_i that has V_{Si} vector, $V_{Si} = (x_1, x_2, x_3, \dots, x_n)$, and x_i is the term weight of the i^{th} term in S_i, and given a sentence S_j that has V_{Sj} vector, $V_{Sj} = (y_1, y_2, y_3, \dots, y_m)$, and y_i is the term weight of the i^{th} term in S_j, and $S_i, S_j \in T$ then, the VSM similarity can be defined by the cosine of the angle between the vectors $\mathbf{V_{Si}}$ ***and*** $\mathbf{V_{Sj}}$:

$$sim(Si, Sj) = cos(V_{Si}, V_{Sj}) = \frac{V_{Si} \cdot V_{Sj}}{|V_{Si}| . |V_{Sj}|} \qquad (6)$$

So, if we represent $\overline{S_i} = (w_{1,i}, w_{2,i}, w_{3,i}, \dots, w_{t,i})$ $\boldsymbol{and}$ $\overline{S_j} = (w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{t,j})$, where $w_{1,i}$ the weight of term 1 in S_i, $w_{1,j}$ is the weight of term 1 in S_j, and t is the number of the terms in the text T, then, the cosine similarity can be rewritten as follow:

$$cos(\overline{s_i}, \overline{s_j}) = \frac{\overline{s_i} . \overline{s_j}}{|\overline{s_i}| . |\overline{s_j}|}$$

$$cos(\overline{s_i}, \overline{s_j}) = \frac{\sum_{n=1}^{t} w_{n,i} \, w_{n,j}}{\sqrt{\sum_{n=1}^{t} w_{n,i}^2} \sqrt{\sum_{n=1}^{t} w_{n,j}^2}} \qquad (7)$$

The VSM performs more statistical investigation than the Jaccard coefficient, and it can distinguish between the important terms that appear in certain domains and the unimportant terms that appear in every text. However, the VSM does not solve the polysemy and synonyms problems and should be supplemented with a semantic analyzer. Our method is supplemented with the LSA similarity analysis that goes beyond the literal existence of the words.

The third level of similarity estimation uses the latent semantic analysis model: in the text mining field, we can see the LSA as a mapping model that transfers the terms-documents matrix, which is sparse and huge, to terms-topics matrix or documents-topics matrix, which is small and informative. The LSA reduces the original matrix to smaller matrix that represents the concepts or the topics mentioned in the text. Thus, the starting point of our process is the original matrix the represents the term-documents relationship. The cells entries reflect the importance of a certain word in a certain document (sentence [10]). The LSA applied the SVD algebraic method to make the necessary factorization analysis to reduce the number of rows and columns. In (Press, Teukolsky, Vetterling, & Flannery., 2007), Press et al. explained the algebraic theory behind the SVD and showed how the SVD decomposes the original matrix to three matrices as shown in Equation 8.

$$X = \cup \Sigma V^T \qquad \boldsymbol{(8)}$$

[10] We concern in this research by the sentence as the unit of text and perform all the similarity assumptions and calculations based on the sentences found in the document, because our deletion process works also at the sentence level to generate the required extract.

Where X is $i \times j$ - the original matrix with rank k, k = min (i, j), U is $i \times i$ matrix that represents the left singular vector, Σ is a diagonal matrix, V is $j \times j$ matrix that represents the right singular vector, and in U, V^T the columns are orthonormal.

As showed in (Golub & Reinsch, 1971) and (Press, Teukolsky, Vetterling, & Flannery., 2007), the mathematical definition of the SVD is complicated, but we simplified it and showed how the SVD can be used to reduce the original matrix of a huge text corpus.

Definition 6. Representing X as a set in a vector space and $X = \{\overline{v}_1, \overline{v}_2, \overline{v}_3, \dots, \overline{v}_n\}$, if $\forall\, \overline{v}_i, \overline{v}_j \in X, i \neq j, \overline{v}_i . \overline{v}_j = 0$, and $\forall v \in X: \|\overline{v}\| = 1$, then the vectors are called orthonormal.

In decomposing the matrix X, we transfer X from high dimensional space of rank k (terms-sentences space) to lower dimensional space of rank r (terms-topic space represented in U and sentences-topic space represented in V) $r < k$. The ∑ diagonal entries represent the singular values σ (the singular value is the square root of the Eigenvalues λ) of X, and they are sorted from largest in $\sum_{1,1}$ to smallest in $\sum_{i,j}$.

Definition 7. Let A be $n \times n$ matrix, λ is called the eigenvalue of A if there is a nonzero vector $\overline{x}$ such *that* $A\overline{x} = \lambda\overline{x}$, $\overline{x}$ is called the eigenvector of A corresponding to λ.

Note that the definition of eigenvalues and eigenvector required $n \times n$ matrix and X is $i \times j$, so we want to obtain a square matrix from X to obtain the eigenvector decomposition.

Lemma 1. Let ***X*** be a $i \times j$ matrix, then the matrix $X.X^T$ is square and symmetric.
Proof.

1. The dimension of X is $i \times j$ and the dimension of $X^T$is $j \times i$ then the dimension of $X.X^T$ will be $i \times i$, this implies that $\mathbf{X.X^T}$is a square matrix.
2. The symmetric means that the transpose of $\mathbf{X.X^T}$ gives the same matrix.

$$(\boldsymbol{XX^T})^{\boldsymbol{T}} = \boldsymbol{X^{T^T}X^T} = \boldsymbol{XX^T} \quad \blacksquare$$

Lemma 2. Let X be a $i \times j$ matrix, then the matrix $X^T.X$ is square and symmetric.
Proof.

1. The dimension of ***X*** is $i \times j$ and the dimension of $X^T$is $j \times i$ then the dimension of $\mathbf{X^TX}$ will be $j \times j$, this implies that $X.X^T$ is a square matrix.
2. The symmetric means that the transpose of X^TX gives the same matrix.

$$(\boldsymbol{X^T X})^T = \boldsymbol{X^T X^{T^T}} = \boldsymbol{X^T X} \quad \blacksquare$$

Now, according to the Definition7 and lemmas 1,2 then we can make eigenvector decomposition, The vectors (columns) in U are eigenvectors of $\mathbf{XX^T}$, and the vectors (columns) in V are the eigenvectors of $\mathrm{X^T X}$ (note that the eigenvalues of $\mathrm{XX^T}$ $\boldsymbol{and}$ $\mathbf{X^T X}$ are the same), so to find the factorization matrices mentioned in equation 8, we follow the following steps:

1. Collect the eigenvalues(λ) of $\mathbf{XX^T}$and the corresponding eigenvectors, normalize the vectors and store them as columns in U (construction of U)
2. Find the square root of **λ**'s and store them in descending order in the diagonal of $\sum$ **=** $(\sigma_1, \sigma_2, \dots, \sigma_k)$ (Construction of $\sum$)
3. Collect the eigenvectors of $\mathbf{X^T X}$, normalize the vectors and store them as columns in $\mathbf{V}$ (construction of $\mathbf{V}$)

The diagonal matrix $\sum$ reflects the strength of the concepts. The $\sum$ is the core of space reduction that LSA performs, the main diagonal of $\sum$ contains the singular values so equation 8 can be seen as:

$$\mathbf{X} = \cup\, \mathbf{diag}(\boldsymbol{\sigma_1, \sigma_2, \dots, \sigma_k})\mathbf{V^T}$$

The reduced SVD performs the required reduced rank approximation and transforms the matrix $\mathbf{X_k}$ with k rank to $\mathbf{X_r}$ approximation matrix with $\mathrm{r} < \mathrm{k}$ by setting the lowest $\mathrm{k} - \mathrm{r}$ eigenvalues in $\sum$ to zeroes, note that the number of concepts equals to the number of $\boldsymbol{\sigma's}$ (singular values) and if some $\boldsymbol{\sigma'}$ values are very small; this means that some of the concepts are not the core of the text (sentence or document), and the LSA will not consider them.

$$\boldsymbol{X_r} = \cup\, \boldsymbol{diag}(\underbrace{\boldsymbol{\sigma_1, \dots, \sigma_r}}_{\text{r}}, \underbrace{\mathbf{0}, \mathbf{0}, \dots \mathbf{0}}_{\text{k-r}})\boldsymbol{V^T}$$

$$\xrightarrow{\boldsymbol{yields}} \boldsymbol{X_r = U_r . S_r . V_r^T} \qquad (\mathbf{9})$$

Where $\mathbf{X_r}$ is **i x j** matrix and represents a reduced rank approximation of the matrix X, $\mathbf{U_r}$ represent the first r columns of U, $\mathbf{S_r}$represent the upper **r x r** of $\sum$, and $\mathbf{V_r^T}$ represent the first r columns of $\mathbf{V^T}$

After constructing the $\mathrm{X_r}$ matrix, the computation of the similarity between any two sentences $\mathrm{S_i}, \mathrm{S_j}$ can be accomplished by computing the dot product of the corresponding columns in $\mathrm{S_r . V_r^T}$ matrix. See Lemma 3.

Lemma 3. Given T as a set of sentences $T = \{S_1, S_2, S_3, \dots, S_M\}$, X is a $i \times j$ term-sentences matrix representing T, and X_r is the reduced SVD of the matrix X. The inner product of the columns vectors $\overline{S_i}, \overline{S_j} \in X, i \neq j$ is the inner product of the corresponding columns in $S_r . V_r^T$

Proof. Let $S_r = S_{rxr}$, $U_r = U_{ixr}$,and $V_r = V_{jxr}$**,** where ***i*** is the number of terms in X, and j is the number of sentences in T, r is the rank of X_r.

$$\boldsymbol{X_r^T X_r} = (\boldsymbol{U_r S_r V_r^T})^T (\boldsymbol{U_r S_r V_r^T}) = \boldsymbol{V_r S_r U_r^T U_r S_r V_r^T} = \boldsymbol{V_r S_r^2 V_r^T} = (\boldsymbol{S_r V_r^T})^T (\boldsymbol{S_r V_r^T})$$

Note that S_r contains zeroes except for the diagonal entries (diagonal matrix) $S_r = S_r^T$, and $U_r . U_r^T = I$ **,** U_r, U_r^T are orthonormal. ■

In summary, LSA transfers the term-document matrix to a term-topic and a sentences-topic matrix that has low dimensional spaces. It uses reduced SVD rank approximation to map semantically related terms or sentences to low dimension space that represents their meaning. After the dimension reduction, the inner product determines the similarity between the vectors of S (using $S_r . V_r^T$matrix) or between terms (using $S_r . U_r$ matrix).

The fourth level of similarity estimation uses the Multi-Layer Similarity: the fourth level of similarity estimation comes to collect the advantages of the three previously mentioned levels. If we consider the previously levels, we can note that each one has significant strengths and weaknesses. The Jaccard coefficient is fast and simple to implement, but it has no means to determine the important terms in the sentence. The VSM computes the similarity based on the term importance, but it faces the synonyms and polysemy problems. The LSA model computes the similarity based on the shared topic (meaning) of the two sentences, but it is hard to implement due to time constraints and the heavy algebraic computations (He, Deng, & Xu, 2006). Therefore, we proposed a new similarity calculation approach that can benefit from the strong points found in each similarity approach discussed in this section and employs the LSA in an efficient way that reduces the number of runs of the LSA extraction.

As we presented in the literature review chapter, the SVD is time-consuming. Thus, the solution we proposed is to reduce the dimensions of the original matrix before running the SVD because the SVD with a huge matrix is an obstacle. The MLS reduction is based on producing a small matrix from the original matrix by removing the parts of the original matrix that can be processed in the first (Jaccard) and second (VSM) levels of the similarity estimations that are discussed previously. The first level will process the sentences that share significant portions of the text, and the second level will process the sentences that have vectors of small angles in the vector space, and the

remaining sentences can then be processed by the SVD to perform the final reduction. Thus, the MLS reduction can allow us to run the latent semantic analysis model over a huge text in a reasonable time interval.

Definition 8. Given $\mathrm{i \times j}$ original matrix X, let $\mathrm{Jacj_{red}}$ be the number of sentences omitted by the Jaccard extraction, and let $\mathrm{VSMj_{red}}$ be the number of sentences omitted by the VSM extraction, then the new j dimension $\mathrm{j_{red}}$ of matrix X will be

$$\boldsymbol{j_{red} = j - (Jacj_{red} + VSMj_{red})}$$

Definition 9. Given $\mathrm{i \times j}$ original matrix X, let $\mathrm{Jaci_{red}}$ be the number of terms omitted by the Jaccard extraction, and let $\mathrm{VSMi_{red}}$ be the number of terms omitted by the VSM extraction, then the new i dimension $\mathrm{i_{red}}$ will be

$$\boldsymbol{i_{red} = i - (Jaci_{red} + VSMi_{red})}$$

Example, For document 22, X contains 691 terms in 45 sentences, $\mathrm{Jaci_{red}} = 182,\ \mathrm{VSMi_{red}} = 276, then\ \mathrm{i_{red}} = 691 - (182 + 276) = 233$ and, $\mathrm{Jacj_{red}} = 11,\ \mathrm{VSMj_{red}} = 17, \mathrm{then\ j_{red}} = 45 - (11 + 17) = 17$, this means the input matrix to the SVD will be $\mathrm{X_{233\ x\ 17}}$ (MLS extraction) instead of $\mathrm{X_{691\ x\ 45}}$ (Classical LSA extraction)**.**

The SVD can be applied over the reduced MLS matrix $\mathrm{X_{i_{red}\ x\ j_{red}}}$, because $0 < \mathrm{i_{red}} \leq \mathrm{i}$ and $\boldsymbol{0 <} \mathrm{j_{red}} \leq \mathrm{j}$, (note that both of $\mathrm{j_{red}}$ and $\mathrm{j_{red}}$ are greater than zero because both the Jaccard similarity extraction and the VSM similarity extraction extracts return at least one sentence from the document). And, the reduced SVD produces $\mathrm{X_q}$ $\boldsymbol{where}$ $\mathrm{q} \leq \mathrm{r}$.

$$\boldsymbol{X_q = U_q . S_q . V_q^T} \qquad (\mathbf{10})$$

The computation of the similarity between any two sentences $\mathrm{S_i, S_j}$ in the MLS extraction approach takes into consideration the similarities that are computed in the Jaccard extraction and the VSM extraction and can be viewed as follows:

$$\boldsymbol{sim(S_i, S_j)} = \begin{cases} \dfrac{|\boldsymbol{S_i} \cap \boldsymbol{S_j}|}{\underset{S_i, S_j \neq \emptyset}{\boldsymbol{min}}(|\boldsymbol{S_i}|, |\boldsymbol{S_j}|)}, & \boldsymbol{S_i, S_j \in X} \\ \dfrac{\overline{\boldsymbol{S_i}} . \overline{\boldsymbol{S_j}}}{|\overline{\boldsymbol{S_i}}| . |\overline{\boldsymbol{S_j}}|} & \boldsymbol{S_i, S_j \in X, if\ Jaccard\ Sim < 0.5} \\ \dfrac{\overline{\boldsymbol{S_i}} . \overline{\boldsymbol{S_j}}}{\|\overline{\boldsymbol{S_i}}\| . \|\overline{\boldsymbol{S_j}}\|} & \boldsymbol{S_i, S_j \in S_q . V_q^T, if\ Jac, VSM\ Sim < 0.5} \end{cases} \qquad (\mathbf{11})$$

As mentioned in (He, Deng, & Xu, 2006), (Wang, Xu, & Craswell, 2013), the complexity of the execution of the SVD in the classical LSA is $\mathbf{O(min(td^2, t^2 d))}$, where $\boldsymbol{t}$ is the number of rows and d is the number of columns. In the MLS

the number of terms (rows) reduced from $\mathbf{i\ to\ i_{red}\ where\ i_{red} \ll i}$ and the number of columns (sentence) reduced from $\mathbf{j\ to\ j_{red}\ where\ j_{red} \ll j}$, this yields a complexity of $\mathbf{O(min(i_{red}j_{red}^{2}\ ,\ i_{red}^{2}j_{red}))}$. The difference between $\mathbf{t}$ and $\mathbf{i_{red}}$ and $\mathbf{d}$ and $\mathbf{j_{red}}$ is significant, for example, for the document 22, the $\mathbf{t}$ value was 691 and $\mathbf{i_{red}}$ value was 233, the $\mathbf{d}$ value was 45, and $\mathbf{j_{red}}$ value was 17.

3.3.2 Algorithm Design and Description

This subsection explains the main algorithm used in the MLS model. The algorithm restructures each document as a set of documents. Each sentence represents a document. Unlike the algorithms implemented in (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (AbdelFattah & Ren, 2009), our algorithm did not use the centrality as a feature that adds weight to the sentence score. We employed more sophisticated statistical and semantic techniques to complete the centrality. Also, the algorithm makes a recursive similarity calculation without user or structure intervention such as the user query (El-Haj & Hammo, 2008), the first sentence (Ghwanmeh S. , Kanaan, Al-Shalabi, & Rabab'ah, Enhanced Algorithm for Extracting the Root of Arabic Words, 2009), and the resemblance to the title (Yeh, Hao-RenKe, Yanga, & Meng, 2005). This deregulation makes the algorithm more flexible to generate unfocused generic extract.

To test our method of text extraction, we build four entirely separated extraction systems; MLSExtractor, LSAExtractor, VSMExtractor, and JacExtractor. The development of the LSAExtractor is performed to compare the accuracy and efficiency of the MLS extraction with the classical LSA extraction, and the development of the JacExtractor and the VSMExtractor is performed to compare the accuracy of the MLS extraction with the text extractions that use the terms overlaps feature or the traditional statistical approached based on term frequency and term distribution. The JacExtractor is based on the Jaccard coefficient to measure the overlapped terms between two sentences. The VSMExtractor is based on a tf.idf scheme to calculate the Cosine Similarity. The LSAExtractor investigates the semantic meaning behind the sentence and extracts the semantically related sentences. The LSAExtractor represents the employment of classical LSA in Text Mining. The MLSExtractor combines the strong points found in the previously mentioned systems in a way that reduces the time required to complete the extraction process. In MLSExtractor, the output similarity computations generated from the first three extraction systems were hired in an enhanced similarity equation (equation11). The new similarity approach considers the verbatim, statistical, and semantic features of the text to determine the resemblance between two

pieces of text. After implementing these systems, we experimented them and collected the results for comparison.

As presented in Figure 3.2, the MLS text extraction method includes three stages: it starts by decomposing the document into sentences and sentences to terms and represents the terms by their stems. Next, we computed the necessary parameters and applied the similarity equations described previously. Finally, we used the deletion process that discards individual sentences based on the similarity calculations computed in the previous stage. In contrary to the bushy path and aggregate similarity, our algorithm considers the high similarities, discards low similarity values, and establishes one to one relationship between each pair of sentences. The following stages detail the main steps that are implemented in our method.

Stage 1: Text preprocessing

The pre-processing stage includes stemming, stopwords removal, punctuations removal, and foreign words removal. Also, it includes the representation of the document's sentences in sets of terms; each set contains the stems of sentence's words. In our experiments, the datasets were taken from the Arabic and English Languages, so we employed the Khoja stemmer to find the stem of the Arabic terms and porter algorithm to find the stem of the English terms. Both of these stemmers are used intensively in the text mining field, and the importance of the employment of the Khoja and porter stemmers appears clearly in accelerating our process.

Stage 2: Similarity Estimation

As shown in Figure 3.2, The similarity estimation of the JacExtractor is based on equation 2, the similarity estimation of the VSMExtractor is based on equation 7, and the similarity estimation of the LSAExtractor is based on the inner product of the vectors of $S_r . V_r^T$ matrix. The similarity estimation is performed between each sentence in the document and all other sentences in the same document. As described above, we used the Jaccard coefficient, the cosine similarity based on tf.idf, and the cosine similarity based on LSA analysis in the similarity computation stage and we applied them in two situations, with and without MLS technique.

Part1: Similarity estimation without the MLS model: In this part, the Jaccard, VSM, and LSA were applied separately, and their results were collected in three different matrices (*JacSim, VSMSim, and LSASim)*. The JacSim matrix collects the results of applying the Jaccard similarity, The VSMSim matrix collects the results of applying the

VSM similarity, and The LSASim matrix collects the results of applying the LSA Similarity. The next Pseudo code represents the similarity estimation without the MLS model :

Assume that $\boldsymbol{d}$ *is a set of sentences composing certain text document,* $\boldsymbol{t}$ *is the number of terms in* $\boldsymbol{d}$*,* $\boldsymbol{n}$ *is the number of sentences,* $\boldsymbol{X_{t\,x\,n}}$ *is our original matrix, and* $\boldsymbol{S_i}$*and* $\boldsymbol{S_j}$ *are any two sentences in* $\boldsymbol{d}$*.*

Input: d, t, n, $X_{t\,x\,n}$
Output: three (n x n) matrices: JacSim, VSMSim, and LSASim.
Process: computing the similarity using the Jaccard, VSM, and LSA models

Construct three (n x n) matrices: JacSim, VSMSim, and LSASim.
For each pair of sentences Si, Sj
Fill JacSim (i , j) using equation 2.
Fill VSMSim (i , j) using equation 7
Fill LSASim (i , j) using equation 9

The output of this stage is three symmetric matrices with main diagonal values equal to one. The document being processed has three similarity matrices, the first is generated from equation 2, the second is generated from equation 7, and the third is generated from equation 9. Each cell represents the similarity value between the two sentences. The procedure detailed above is greedy and the required complexity -in terms of space and time- is high because of the intensive runs of the LSAExtractor (the LSA procedure will be executed between any two sentences in the document). Therefore, we suggested the MLS methods in part2 of our experiment.

Part2: Similarity estimation with the MLS model: Figure 3.2, shows the structure of the MLS extraction, after making the pre-processing stage, the system creates a terms-sentences matrix $\mathrm{X_{i\,x\,j}}$, and this matrix will be the input for the JacExtractor and will be processed using equation 2, and to VSMExtractor and will be processed using equation 7, and to the SVD factorization subsystem. The SVD factorization subsystem reduces the matrix dimensions and produces $\mathrm{X_r}$ which will be the input of the LSAExtractor. At this moment, three extracts will be generated for each document; one from each extractor (the dashed line in Figure 3.2), these extracts will be used later in our research to construct the inverted index for the IR system.

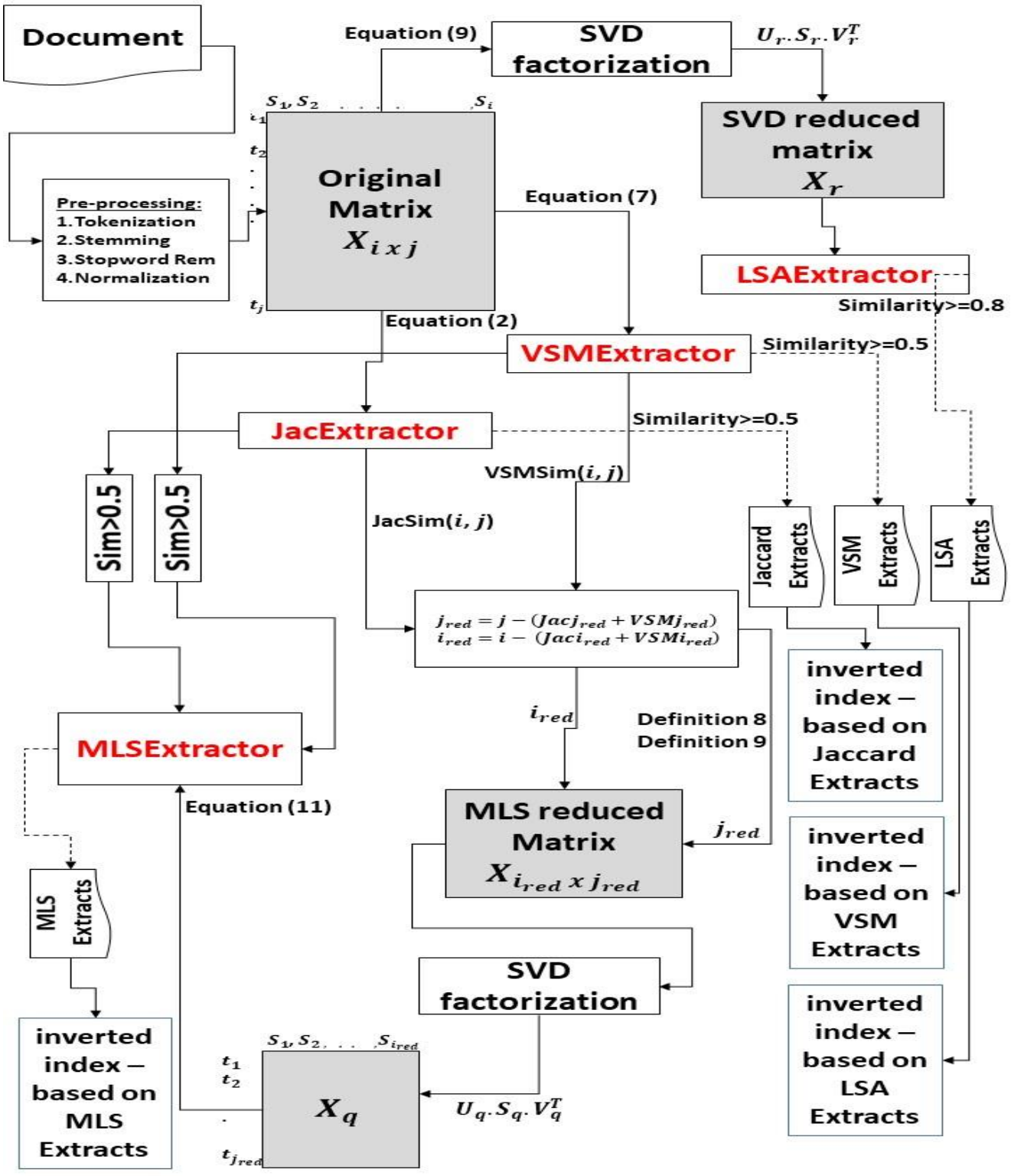


Figure 3.2 MLSExtractor Architecture

Another two outputs that are generated from the JacExtractor and the VSMExtractor are:

1. The similarity values greater than 50% that go directly as input to the MLSExtractor similarity matrix, here we start from Jaccard, and then we move to the VSM, so the Jaccard will be the first similarity estimation that should be performed (with and without MLS extraction).

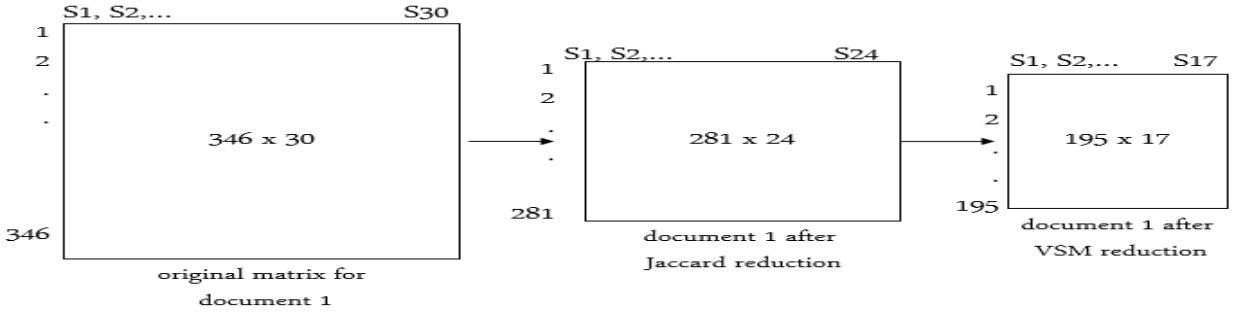
2. The sentences omitted by JacExtractor ($\mathbf{Jacj_{red}\ and\ Jaci_{red}}$) and VSMExtractor ($\mathbf{VSMj_{red}\ and\ VSMi_{red}}$). These parameters help to identify the new dimensions of the original matrix (i_{red} and j_{red}) by applying definition 8 and definition 9 equations. The values of i_{red} and j_{red} represent the amount of reduction obtained by the MLS hierarchy. The output at this point is the matrix X_q that ***is smaller than the original matrix X by*** i_{red} sentences and j_{red} terms

Then, the reduced MLS matrix X_q is processed by the SVD factorization system and equation 11 is applied to generate the similarity matrix for the MLSExtractor system. AT this point, the SVD will process small matrix comparing with the matrix that the first SVD system process (the SVD in the LSAExtractor). The MLSExtractor receives the similarity values coming from three sources, the JacExtractor (the sentences with Jaccard similarity greater than 50%), the VSMExtractor (the sentences with VSM similarity greater than 50% and not in the list provided by the JacExtyractor), and the SVD factorization subsystem (this represents the application of equation 11) and merges them to produce a single similarity matrix.

In abstract words, the method starts by computing the Jaccard similarity and decide whether the VSM and the LSA similarity computations are necessary or not. If the Jaccard similarity between two sentences is high (greater than 50%), this implies that the two sentences shared a sufficient portion of text and no further calculations are needed. The high value of the Jaccard similarity can omit the VSM and the LSA similarity processing for a considerable number of sentences. For example, the Jaccard between the following pairs of sentences from document1 exceeded 50% : ((S_1, S_4), (S_1, S_9), (S_1, S_{10}), (S_5, S_6), (S_5, S_8), (S_8, S_{11}), (S_9, S_{16}), (S_{23}, S_{27})). So, no further similarity computation is required for those pairs of sentences, and all the LSA similarity computation of the sentences S_4, S_6, S_8, S_9, S_{10}, and S_{27} will be omitted. Note that from the pair (S_1, S_4), we save 26 runs of the LSA procedure because all the LSA similarity of the pairs

(S_4, S_5), (S_4, S_6),…, (S_4, S_{30}) will be discarded. (the number of sentence in document1 is 30)

Document 1 in Essex corpus contains 30 sentence and 346 distinctive terms, the dimensions of the original matrix X is 30x346, and after the MLS reduction, these dimensions became 17x195, the difference between i and i_{red} is 13 (43% reduction) and the difference between the j and j_{red} is 151 (44% reduction). Also, the number of runs of the Si, Sj similarity computations reduced from 435 in classical LSA extraction to 241 in MLS extraction (reduced the number of runs of the LSA similarity by 45%). The form of the matrix reduction performed looks like the following:



In the MLS extraction the previous pseudo code is amended as follow:

Assume that $\boldsymbol{d}$ is a set of sentences composing certain text document, $\boldsymbol{t}$ is the number of terms in $\boldsymbol{d}$, $\boldsymbol{n}$ is the number of sentences, $\boldsymbol{X_{t\,x\,n}}$ is our original matrix, and $\boldsymbol{S_i}$and $\boldsymbol{S_j}$ are any two sentences in $\boldsymbol{d}$.

Input: d, t, n, $X_{t\,x\,n}$
Output: two (n x n) matrices: JacSim, MLSim.
Process: computing the similarity using the MLS model

Construct two (n x n) matrices: JacSim, MLSim
for each pair of sentences $\boldsymbol{S_i}$and $\boldsymbol{S_j}$ in $\boldsymbol{X_{t\,x\,n}}$
fill JacSim(i,j) using equation 2.' The Jaccard will be the first step with or without MLS
$\boldsymbol{i_{red} = t}$, $\boldsymbol{j_{red} = n}$
for each entry in JacSim matrix
if JacSim(i,j)>50%
MLSim(i,j)=JacSim(i,j) and Delete $\boldsymbol{S_j}$ column and rows(for each t in $\boldsymbol{S_j}$) from $\boldsymbol{X_{t\,x\,n}}$
$\boldsymbol{i_{red} = i_{red} - number\ of terms\ in\ S_j}$, $\boldsymbol{j_{red} = j_{red} - 1}$
run the VSMExtractor over the $\boldsymbol{X_{t\,x\,n}}$,
for each $\boldsymbol{S_i}$and $\boldsymbol{S_j}$ not found in MLsim(i,j)
If the VSM Similarity (i, j)>50%
MLsim(i,j)=VSM(i.j) and Delete $\boldsymbol{S_j}$ column and rows(for each t in $\boldsymbol{S_j}$) from $\boldsymbol{X_{t\,x\,n}}$
$\boldsymbol{i_{red} = i_{red} - number\ of terms\ in\ S_j}$, $\boldsymbol{j_{red} = j_{red} - 1}$
Construct new terms-documents matrix $\boldsymbol{X_{i_{red}\,x\,j_{red}}}$
run the LSAExtractor over $\boldsymbol{X_{i_{red}\,x\,j_{red}}}$
MLsim(i,j) = LSASim(i,j)

This pseudo-code implements definition 8 and 9 to obtain the reduced matrix and equation 11 to construct the similarity matrix. From the pseudo-code, any Jaccard similarity greater than 50% is considered significant, and this similarity value is stored in the MLSSim matrix, and the corresponding columns in the original matrix are deleted. In a similar manner and after finishing the Jaccard processing, each time the algorithm finds a large value of the VSM cosine similarity the corresponding column and rows of the similar sentence are removed from the original matrix $X_{t\,x\,n}$. After the Jaccard and VSM processing, the MLS pseudo-code applies the LSAExtractor at the end of this algorithm, the original matrix is reduced to $X_{i_{red}\,x\,j_{red}}$. It appears clear that the part of the algorithm that consumes a lot of time is shifted to the end and used with a small number of input data (terms and sentences).

Stage 3: The Deletion Process

The deletion algorithm is the core of the MLS and it is used to delete the repetitive sentences after the generation of similarity matrices. It is a recursive procedure that investigates the diversity among sentences and removes the sentences that have similar sentences in the text. The algorithm gives all sentences found in the text being summarized the same value of importance, and the only condition for sentences discarding is the centrality of the sentence relative to the other sentences found in the text. Therefore, our algorithm can be applied to one document, multi-document, or any piece of text that contains two sentences or more without bias to structural, linguistic, or domain features.

Two parameters are considered during the deletion process; the similarity values and the existence of a base sentence. The parameters identification, the threshold value of the similarity values, and the deletion process design are the three main steps in the deletion process explanation:

Step 1: Parameters Identification

In this step, we determine the conditions and parameters that are necessary to delete the repetitive sentences. The first parameter used in the deletion process is the similarity values between the sentences that are generated in the second stage by the Jaccard, VSM, LSA, and MLS extractors. The similarity value between two sentences decides whether one of the sentences will be deleted or not if this similarity is significant. The second parameter that the deletion process considered is the existence of a base sentence S_i. The system deletes S_j if the similarity between

S_i and S_j exceeded the threshold value and S_i was not deleted before; otherwise, S_j remains. We cannot remove S_j if the similar sentence S_i was already removed. So, the final condition that controls the deletion process is:

$$\boldsymbol{if\ the\ Similarity\ between\ (S_i, S_j) \geq threshold\ value\ and\ S_i \notin Deleted\ list, then\ delete\ S_j}$$

Step 2: Threshold of the Similarity Value

The deletion process would delete the repetitive sentences if their similarity with previously mentioned sentence exceeded certain boundaries. In the subsection, we explain our experiment that determines the similarity threshold value. In this regard, two parameters should be formally defined: the Ratio of Sentences Intersection (RSI) and the condensation rate.

Definition 10: Let A = {S_1, S_2… S_n} be a set of sentences in automatic extract generated by one of our automatic extractor systems for document d_i. And let M = {S_1, S_2… S_m} be a set of sentences in the reference extract for document d_i, then

$$\boldsymbol{RSI(A, M) = \left(\frac{|A \cap M|}{min(n, m)}\right)(100\%) \ldots . (12)}$$

Example: the fourth manual extract (M4) of document 1 contains the sentences 2, 5, 7, 8, 13, 15, 17, 29 and the MLSExtractor automatic extract of document 1 contains the sentences 1, 2, 5, 7, 14, 15, 17, 18, 20, 22, 26, and 29.

$$\boldsymbol{RSI\ (M4\ , MLS\ extract) = \left(\frac{6}{8}\right)(100\%) = 75\%}$$

In this example, 75% of M4 sentences are appeared in the MLSExtractor extract (for document1). The RSI value ranges between 0 and 100%. The value 100% of the RSI means that all the reference summary sentences are found in the automatic extract.

Definition 11: Let t be the number of terms in document d, and t1 is the number of terms in the extract e:

$$\boldsymbol{CR(e) = \left(\frac{t1}{t}\right)(100\%) \ldots (13)}$$

Example: document1 contains 414 terms and MLSExtractor extract contains 150 term, the $\mathrm{CR} = \left(\frac{150}{414}\right)(100\%) = 36\%$. The CR value ranges between 0 and 100%. The value 100% for the CR means no condensation occurred.

After defining the RSI and CR parameters, we can explain how we chose our threshold values. To specify the threshold values of the similarity values, we generated the similarity matrices for a sample of 13 documents. The selected documents have a variance number of sentences, for example, document 1 contains 30 sentences,

document 2 contains 15 sentences, and document 3 contains 4. The documents are manipulated by the three extractors JacExtractor, VSMExtractor, and LSAExtractor and three similarity matrices were generated for each document.

The first look at the similarity matrices of the documents in our sample gave us an indication of the possible threshold values. For VSMExtractor and JacExtractor, we tested three possible ranges: greater than 25%, greater than 50%, and greater than 75%. We computed the RSI and CR at these three values, and we found that the value greater than 25% generated insignificant RSI (40% for the Jaccard and 36% for the VSM). The value greater than 75% produced a significant value of RSI (84% for the Jaccard and 99% for the VSM), but the CR was insignificant (81% for the Jaccard and 99% for the VSM). The value greater than 50% generated significant values in both RSI (67% for the Jaccard and 84% for the VSM) and CR (61% for the Jaccard and 81% for the VSM), Figure 3.3.a and 3.3.b presented the results.

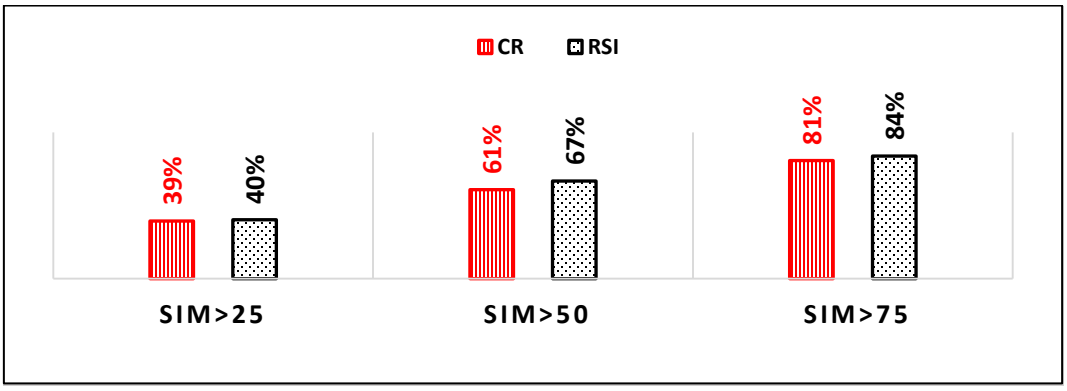


Figure 3.3.a Condensation Rate with RSI at 25%, 50%, and 75% of the Jaccard similarity.

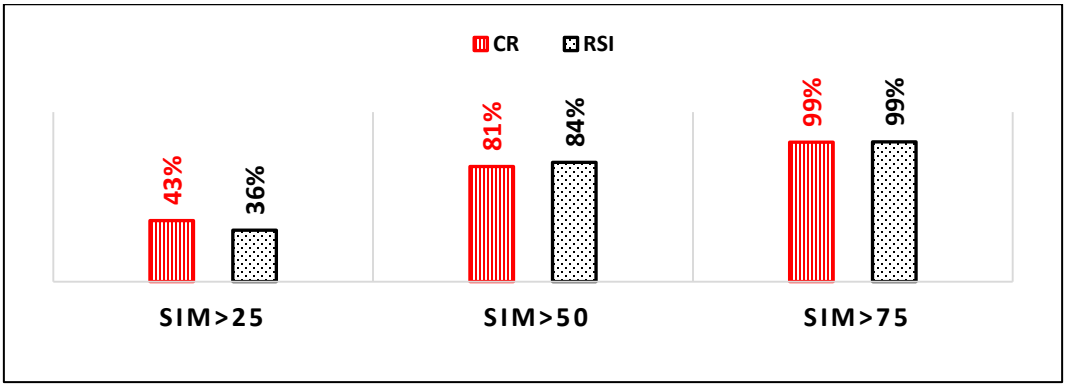


Figure 3.3.b Condensation Rate with RSI at 25%, 50%, and 75% of the VSM similarity.

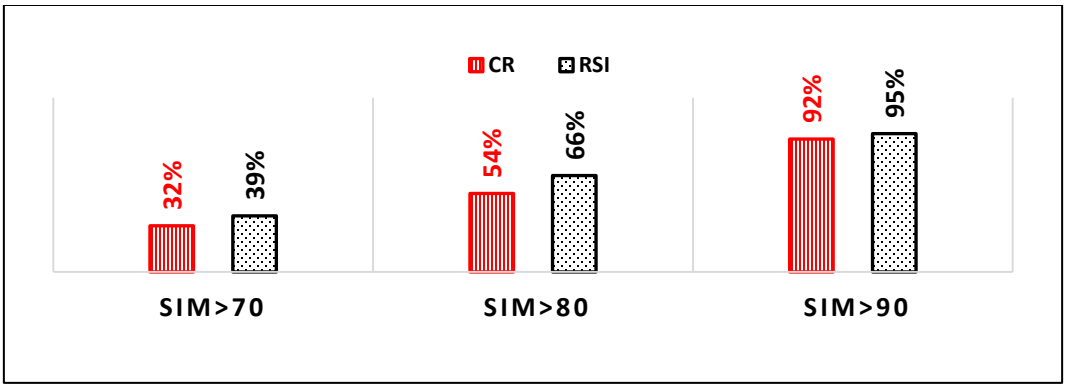


Figure 3.3.c Condensation Rate with RSI at 70%, 80%, and 90% of the LSA similarity.

For LSAExtractor, the generated similarities between the sentences were high because the document is talking about one topic or related topics. Therefore, we increased the tested values to be: greater than 70%, greater than 80%, and greater than 90%. The RSI and CR values were computed and presented in Figure 3.3.c; the most significant values of the RSI and CR were obtained on greater than 80% threshold value.

Step 3: Deletion Process design

The difference between the four extractors, which are developed in this research, comes in the similarity estimation not in how we delete the repetitive sentences. Each extractor generates a similarity matrix, and this matrix is the input to the deletion process. The deletion process is impeded in our extractors in a similar manner, and it processes the similarity matrices to decide which sentences should be discarded.

The employment of the deletion process in the JacExtractor, the VSMExtractor, and the LSAExtractor is straightforward. The deletion process examines the similarity matrix entries against the threshold values (50% for the JacExtractor andVSMExtractor and 80% for the LSAExtractor) and omits the sentences based on the condition mentioned in step1. But the implementation of the deletion process in the MLSExtractor is different because the MLSEXtractor similarity matrix collects the similarity values from three sources; the Jaccard similarities in the first layer, the VSM similarities in the second layer, and the LSA similarities in the third layer (see Figure 3.2). Therefore, parts of the MLSEXtractor matrix should be examined against 50% value (first and second layers similarities), and the remaining parts should be examined against 80% value (third layer similarities). The layer number should be reflected in the deletion process. We restructured the similarity matrix, and each entry is implemented as a pair of number and identifier, the number represents the similarity value, and the identifier takes a binary value, 0 indicates

that the similarity value came from the first or second layer, and 1 indicates that the similarity value came from the third layer similarity.

The pseudo-code of the deletion algorithm in the MLS extraction is as follow:

Input: MLSim (j x j); where j is the number of sentences
Output: list of extract sentences id's E, a list of deleted sentences id's $\boldsymbol{Del}$
Process: deletion process

for each pair of sentences $\boldsymbol{S_i}$ *and* $\boldsymbol{S_j}$ *in*
if $i \neq j$ *then*
if j.identifier = 0
{if MLSim (i x j)>0.5 and $\boldsymbol{i} \notin \boldsymbol{Del}$
add i to E and add j to $\boldsymbol{Del}$*}*
else if MLSim (i x j)>0.8 and $\boldsymbol{i} \notin \boldsymbol{Del}$
add i to E and add j to $\boldsymbol{Del}$

Note that after applying the Jaccard similarity between two sentences (S_i, S_j) and finding that more than half of the terms are shared, then the MLSExtractor deletes the second sentence S_j and all the LSA similarity calculations required for the S_j sentence will be canceled. For example, in document 1 that contains 30 sentences, the Jaccard similarity between S_1 and S_4 is 67%. The MLSExtractor deletes S_4. The deletion of S_4 omitted 25 runs of the LSA similarity procedure runs because it cancels the similarity computations of S_4 with the sentences S_5 to S_{30}. In the same way, the Jaccard similarity between (S_1 and S_9) is 54%. S_9 will be deleted. The deletion of S_9 omitted 20 runs of the LSA similarity procedure runs because it cancels the similarity computations of S_9 with the sentences S_{10} to S_{30}. For d_1, the MLS similarity computations reduced the number of runs of the LSA procedure from 415 to 241, and the matrix that the LSA should decompose reduced from 346 *x* 30 to 195 *x* 17.

The input of the deletion algorithm will be $X_{j\,x\,j}$ matrix where j is the number of sentences. $X_{j\,x\,j}$ has $\left(\frac{j\,x\,j}{2}\right)$ elements (and$\left(\frac{j\,x\,j}{2}\right)$ are empty) and the diagonal values are ones represents $\mathrm{sim}(S_i, S_i)$**.** This matrix generated from the Jaccard, VSM, LSA, or MLS similarity approach. Each matrix entry represents the similarity value between two sentences, and the algorithm should decide whether one of them should be omitted from the final extract or not. The algorithm starts with the first sentence and recognizes its similarity with all other sentences. With n sentences, the deletion process takes n-1 comparisons to process S_1, n-2 to process S_2, n-3 to process S_3, and so on until it reaches the $S_{n-1,}$ which takes one comparison. The time complexity of this process can be depicted as:

$$\boldsymbol{Number\ of\ comparisons} = (\boldsymbol{n-1}) + (\boldsymbol{n-2}) + \cdots + \boldsymbol{1}$$

$$= \frac{\boldsymbol{n}\,(\boldsymbol{n}-\mathbf{1})}{\mathbf{2}} = \frac{\boldsymbol{n^2}}{\mathbf{2}} - \frac{\boldsymbol{n}}{\mathbf{2}} = \frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{n^2}-\boldsymbol{n})$$

Accordingly, the deletion process takes O($\mathbf{n^2}$) to process n sentences. Normally, the number of sentences in the document is not large, especially in the single-document summarization, so the deletion process complexity is not a repellent factor.

3.4 NBDV Synonyms Extraction Method

The NBDV is developed in this investigation. It is a vector space-based synonyms extraction method that considers three aspects during the synonyms extraction process:

1. Making the NBDV completely statistical and this means that during all the phases of extraction, the NBDV does not use a database of stored synonyms, meanings, or patterns.
2. Processing the nouns as meaningful units, not as a bag of words (as in the CBoW and SG models) and capturing the noun's meaning by precisely collecting the verbs that are specific to a group of nouns (the parameters used to identify the verb uniqueness is described in section 3.4.1).
3. Reducing the problem domain, the developed weighting scheme weights the parts of the corpus that are related to the noun being processed.

The NBDV method uses unsupervised learning to extract nouns synonyms. Definition 12 gives a simple interpretation of the NBDV method:

Definition 12:
Given $\mathbf{S_{n1}}$ and $\mathbf{S_{n2}}$ (Verb_Noun adjacent lists) are the sets of verbs adjoin the nouns $\mathbf{n_1}$ and $\mathbf{n_2}$ such that:
$\mathbf{S_{n1}} = \{\mathbf{n_1v_1}, \mathbf{n_1v_2}, \mathbf{n_1v_3}, \dots, \mathbf{n_1v_i}\}$,
$\mathbf{S_{n2}} = \{\mathbf{n_2v_1}, \mathbf{n_2v_2}, \mathbf{n_2v_3}, \dots, \mathbf{n_2v_j}\}$.

Where $\mathbf{n_1v_1}$ is the first verb adjacent to the noun $\mathbf{n_1}$ and $\mathbf{n_2v_1}$ is the first verb adjacent to the noun $\mathbf{n_2}$. Both $\mathbf{i}$ and $\mathbf{j}$ are positive integers greater than 1.

We said that $\mathbf{n_1}$ and $\mathbf{n_2}$ are synonyms if $|\mathbf{S_{n1}} \cap \mathbf{S_{n2}}| \geq \mathbf{c}$ (Threshold value)

Example: if $\mathbf{n_1}$ is "سيارة" (car) and $\mathbf{n_2}$ is "مركبة" (vehicle).

$$\mathbf{S}_{\text{سيارة}} = \begin{Bmatrix} \text{تسير, ينتج, يشتري, يقلع, يصتدم, يركن, يركب, يصلح, يقود} \\ \text{يبيع, يدهس, يسلك, ...} \end{Bmatrix}$$

$$\mathbf{S}_{\mathbf{car}} = \begin{Bmatrix} \mathbf{drive}, \mathbf{repair}, \mathbf{ride}, \mathbf{park}, \mathbf{bump}, \mathbf{take\ off}, \text{buy}, \\ \mathbf{produce}, \text{walk}, \text{walk}, \text{tread}, \text{sell}, \ldots \end{Bmatrix}$$

$$\mathbf{S}_{\text{مركبة}} = \begin{Bmatrix} \text{ينتج, يقلع, يحلق, يبحر, يطير, يتوقف, يصتدم, يركب, يصلح, يقود} \\ \text{يسقط, يغرق, يرسو, يهبط, يركن ...} \end{Bmatrix}$$

$$\mathbf{S}_{\mathbf{vehicle}} = \begin{Bmatrix} \mathbf{drive}, \mathbf{repair}, \mathbf{ride}, \mathbf{bump}, \text{fly}, \text{sail}, \mathbf{take\ off}, \text{crash}, \\ \mathbf{produce}, \mathbf{park}, \text{land}, \text{land}, \text{sink}, \text{fall} \ldots \end{Bmatrix}$$

If c = 5 then سيارة and مركبة are synonyms because: $\left|\mathbf{S}_{\text{سيارة}} \cap \mathbf{S}_{\text{مركبة}}\right| = \mathbf{7} > \mathbf{c}$

Definition 12 presents the main idea; it considers two nouns as synonyms if they share more than *c* verbs. The verbs such as “buy”, “produce” “walk” are general verbs and can be found with a wide range of nouns, so they cannot be used as distinguishing verbs. Thus, the selection of the verbs that can group the nouns to semantically related groups is more complicated. Section 3.3.1 depicts the criteria that have been used to select the distinctive verbs and explains how these criteria have been employed to weight the verbs in the proposed NBDV method.

Two aspects should be mentioned before proceeding to detail the NBDV model.

- The nouns targeted by the NBDV method are the common nouns, not the proper or the entity nouns because the latter mostly do not have synonyms.
- The NBDV model can be seen as a synonyms extraction, or more generally, as a collector of semantically related words because it combines the terms that normally share one semantic context. This point is important to mention because in the evaluation process of the relevance of the output set, the evaluator should not make an exact match between the automatically generated set and the answer set that is taken from the base dictionary. In addition, considering the NBDV as a semantic word collection model makes it more supportive to the other fields of information systems such as query expansion in the Information Retrieval systems.

The NBDV method includes two phases, the weighing phase that uses the OWS scheme, and the synonyms detection phase that uses the cosine similarity between the vectors that generated from the first phase to decide if two nouns are synonyms or not.

3.4.1 The Orbit Weighting Phase

The OWS is used in the weighting phase of the NBDV method to replace the traditional tf.idf weighting scheme used in the skip-gram model or the Continuous Bag-of-Words model (Mikolov, Chen, Corrado, & Dean, 2013). The OWS is designed for nouns because the nouns are the primary concern of the text mining applications, mostly, all the query terms in Information retrieval, the class and category names in text categorization, the concept/entity in entity recognition, and others are nouns.

In the OWS, the nouns that should be processed have semantic relation with the noun that the user wants to find its synonyms. In each run, the similarities are computed between the nouns that share distinctive verbs. The nouns that share a set of specific verbs are more likely to be synonyms. For example, the noun car and automobile have some special verbs that distinguish them from the other nouns such as the verbs “park” “crash”, “drive”.

The reason for selecting the verbs as distinguishing factors is that the other parts of speech are normally used with a wide range of nouns. For example, the adjectives are used in the languages to describe objects with different domains.

The OWS weights the verbs based on their singularity to a group of nouns. In comparison with traditional methods such as the skip-gram or the Continuous Bag-of-Words models, the singularity of a specific verb is determined using three parameters, (1) the number of times the verb appeared with the noun, (2) the number of nouns the verb appeared-with in the corpus, and (3) the average distance between the verb and the noun in each occurrence of verb and noun together. These parameters are necessary to measure the uniqueness of the verb with respect to a specific set of nouns.

The purpose of combining the three parameters is to neglect the verbs that appear with a wide range of nouns. For example, if the verb appeared with a large number of nouns, this implies that the verb is a general verb and the value of the second parameter will be very low (the second parameter is inversely proportional with the number of

nouns the verb appeared-with), and the OWS gives the verb tiny weight. Figure 3.4 depicts the idea behind the OWS weighting. The relations of noun-verbs are represented in an orbiting space in which the noun placed in the center and the verbs round in orbits. The verbs placed in one orbit have the same degree of importance (roughly the same weight value). Also, the verbs with high weight values are spun in the inner orbits, and the verbs with low weight values are confined in the outer orbits.

In Figure 3.4, the verbs v_1 and v_2 (could be the verbs "يقود drive" and "يركن park") appeared in inner orbit, and their contribution will be greater than the contribution of the verbs v_8 and v_9 (could be the verbs " يشتري buy" and "يبيع sell") that are shifted to the outer orbit.

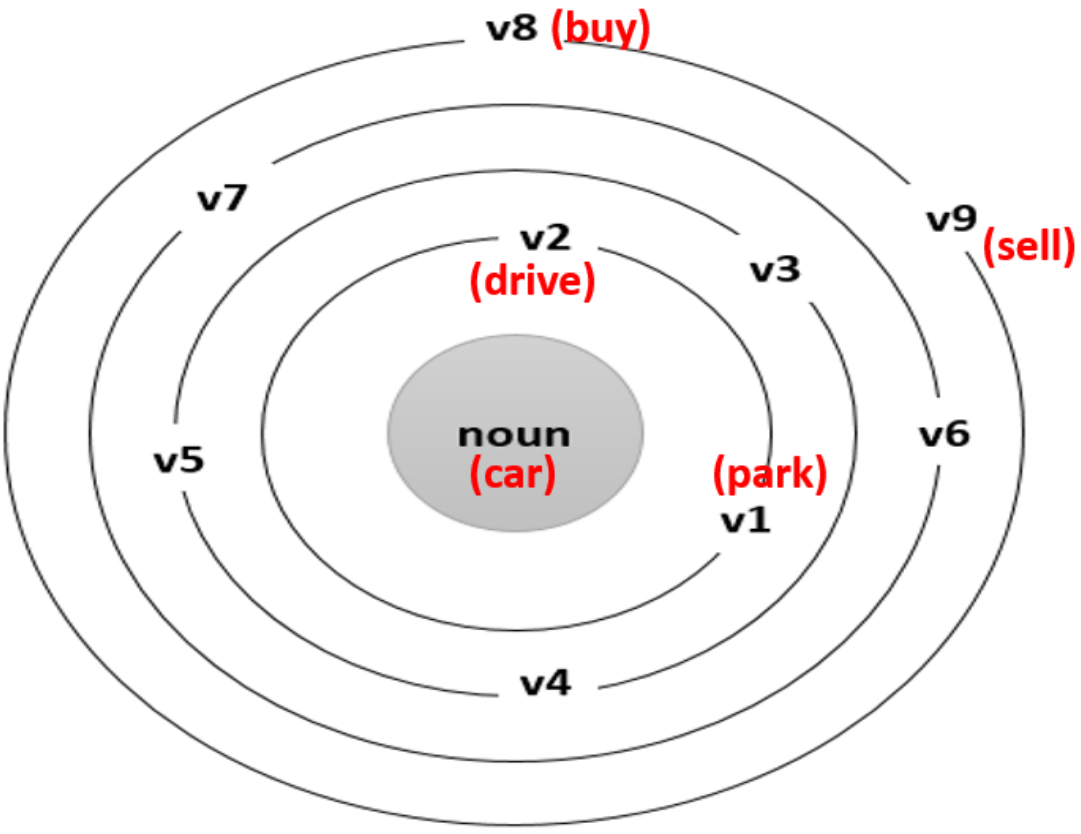


Figure 3.4 Orbit Representation for the Noun-Verbs Relationships

In the OWS, the vector of the noun n is a set of weights of the verbs appeared with n in the corpus, and it can be represented as follow:

$$\vec{n} = (w_{v1}, w_{v2}, w_{v3}, \dots, w_{vi})$$

Where $\mathbf{w_{v1}}$ is the weight of the verb v_1 with respect to n, and i is the number of verbs appeared with n in the corpus.

The weights composing in the noun vector $\vec{n}$ is computing by the OWS weighting scheme. The OWS computes the weight of a specific verb and decides whether this verb belongs to $\vec{n}$, or not.

3.4.1.1 The OWS Identifiers

In Definition 12, two nouns are synonyms if they shared a certain number of verbs. However, our assumption should process the verbs according to their contributions in the text. For example, consider the verb "يشتري" (buy) and the verb "يركن" (park) both of them appeared in example 1 as verbs adjacent to the nouns "مركبة"،"سيارة" , the verb "يشتري" is a general verb that can be used with a lot of objects or services , but the verb يركن is related to fewer objects such as car or bus. The verb "يركن" should make more contribution than the verb "يشتري" in determining that "مركبة" and "سيارة" are semantically related words. It is crucial to determine which verbs should be considered as distinguishing verbs and have a significant effect in determining the synonyms. Therefore, definition 12 can be updated as follow:

Definition 13: Given $\mathbf{S_n}$ as the set of verbs adjacent to the noun n such that: $\mathbf{S_n} = \{\mathbf{n_{v1}}, \mathbf{n_{v2}}, \mathbf{n_{v3}}, \dots, \mathbf{n_{vi}}\}$, Then, for each Verb $\mathbf{v}$ adjacent n, the weight of v is determined by considering the following parameters.

1. VerbNoun Frequency: The number of times the verb v appeared-with the noun n in the whole corpus ($\mathrm{fr}_{(\mathrm{V|n})}$).
2. VerbNouns Distribution: The number of nouns appeared with verb v in the whole corpus ($\mathrm{idf}_{(\mathrm{n|V})}$).
3. VerbNouns Distance: The average distance between the verb v and the noun n in all the (v, n) occurrences ($\mathrm{AD}_{(\mathrm{V|n})}$).

In the NBDV method, $\mathbf{n_j}$ is considered as a synonym to noun n if the similarity between $\mathbf{n_j}$ and n exceeded a certain threshold value, and the similarity is computed based on the weights of the shared verbs that are weighted by considering the three parameters, the $\mathbf{fr}_{(\mathbf{V|n})}$, the $\mathbf{idf}_{(\mathbf{n|V})}$, and the $\mathbf{AD}_{(\mathbf{V|n})}$. These three parameters are the identifiers of the OWS.

Based In definition 13, the verb that frequently appeared with a certain noun (large value of $\mathrm{fr}_{(\mathrm{V}|\mathrm{n})}$), and normally located as close as possible to that noun (small value $\mathrm{AD}_{(\mathrm{V}|\mathrm{n})}$), and appeared with a small number of nouns (small value of $\mathrm{idf}_{(\mathrm{V}|\mathrm{n})}$), obtained high weight value and should be placed in the inner orbits of Figure 3.4.

To define the three parameters, assume that t refers to any term belongs to the dataset K, n refers to any noun belongs to K, v refers to any verb belongs to K, N is the number of nouns in K.

Parameter 1: VerbNoun Frequency ($\mathbf{fr}_{(\mathbf{v}|\mathbf{n})}$)

VerbNoun Frequency is the number of times the verb v and the noun n mentioned together within i^{th} positions. Parameter 1 or the $\mathbf{fr}_{(\mathbf{v}|\mathbf{n})}$ identifies the verbs that commonly appear with a specific noun. The $\mathbf{fr}_{(\mathbf{v}|\mathbf{n})}$ is computed as follow:

$$\boldsymbol{fr}_{(\boldsymbol{v}|\boldsymbol{n})} = \sum_{\boldsymbol{t} \in \boldsymbol{K}} \boldsymbol{count}(\boldsymbol{v}, \boldsymbol{n}\ \boldsymbol{combinations})$$

But, some verbs are general and appear intensively, and others are specific and appear in certain domains and platforms. The normalization of the VerbNoun Frequency is performed by dividing the $\mathbf{fr}_{(\mathbf{v}|\mathbf{n})}$ by the total number of time the verb *v* appeared in the whole corpus:

$$\boldsymbol{fr}_{(\boldsymbol{v}|\boldsymbol{n})} = \frac{\sum_{\boldsymbol{t} \in \boldsymbol{K}} \boldsymbol{count}(\boldsymbol{v}, \boldsymbol{n}\ \boldsymbol{combinations})}{\sum_{\boldsymbol{t} \in \boldsymbol{K}} \boldsymbol{count}(\boldsymbol{v})} \ldots (\mathbf{14})$$

The normalization degrades the importance of the general verbs because the denominator in equation 14 will be high for such verbs.

Parameter 2: VerbNouns Distribution ($\mathbf{idf}_{(\mathbf{n}|\mathbf{v})}$)

VerbNouns Distribution is the number of nouns the verb v appeared-with in the whole corpus.

$$\boldsymbol{idf}_{(\boldsymbol{n}|\boldsymbol{v})} = \sum_{\boldsymbol{t} \in \boldsymbol{K}} \boldsymbol{count\ n\ appeared\ with\ the\ v} \ldots (\mathbf{15})$$

VerbNouns Distribution indicates the singularity of the verb. The large value of the $\mathbf{idf}_{(\mathbf{n}|\mathbf{v})}$ parameter means that the verb can be used with a large number of nouns in multiple domains and situations. If the verb distribution among

the nouns was high, the effectiveness of the verb in differentiating the nouns will be reduced. We argued at this point that the verbs that appear intensively in the text should be treated as Stopwords because for mathematical computation, they will not add any value. The smaller the value of $\mathbf{idf_{(v|n)}}$ the larger the contribution of v. for normalization, equation 15 is rewritten as follows:

$$\boldsymbol{idf_{(n|v)} = log \frac{N}{\sum_{t \in K} number\ of\ nouns\ appeared\ with\ v}} \ldots (16)$$

In IR and NLP, the idf is used to measure the distribution of the term over the whole documents that compose the corpus (Chen & Chiu, 2011). If the idf value was high, this implies that the term appeared in a large number of documents, and the weight of the term will be low. In our weighting scheme, the idf is represented effectively, and it is used to measure the distribution of the verb over all the nouns found in the corpus. Similar to idf used in IR and NLP applications, if the idf value was high, this implies that the verb appeared with a large number of nouns and the verb will not be beneficial in distinguishing the nouns (the weight of the verb will be low).

Parameter 3: VerbNouns Distance ($\mathbf{AD_{(v|n)}}$)

VerbNouns Distance is the average distance between the verb v and the noun n in all occurrences of (v, n) combination.

$$\boldsymbol{AD_{(v|n)} = \frac{1}{Avg|differences\ between\ v\ position\ and\ n\ position\ for\ all\ v,n\ occurrences|}} \ldots (17)$$

In the OWS scheme, the verb and noun are not necessary to be adjacent because, in some cases, certain words (such as Adjectives, adverbs, …) may come between them. For example, consider the following statements from our corpus:

هو يقود السيارة	He drives the car
يقود السائق السيارة	The driver drives the car
يقود السائق المتهور السيارة	The reckless driver drives the car
يقود السائق المخمور برعونة السيارة	Recklessly, the drunk driver drives the car

However, the distance parameter ($\mathbf{AD_{(v|n)}}$) imposes that the effect of the verb on the noun becomes stronger if the verb was adjacent to the noun and this effect reduced as the noun becomes far away from the verb. In the above

example, in sentence number one, the whole concentration will be on the noun car, but in the fourth sentence, it addresses the driver, not the car. So, the $\mathbf{AD_{(V|n)}}$ identifier gives the verb drive with respect to the noun car in the first sentence heavier weight than the verb drive in the fourth sentence with respect to the same noun. The importance of $\mathbf{AD_{(V|n)}}$ parameter lies in showing how close the verb to the noun and normally the adjacency means a robust relationship between the verb and noun. So, the smaller the value $\mathbf{AD_{(V|n)}}$, is the larger the weight of v.

3.4.1.2 The Weighting Equation of OWS

After computing the value of each parameter, the OWS weighting schemes weights of the verb v with respect to noun n as shown in equation 18:

$$\boldsymbol{Weight(v|n) = fr_{(v|n)} * idf_{(n|v)} * AD_{(v|n)} \ldots (18)}$$

Equation 18 summarized the OWS weighting scheme, where $\mathbf{fr_{(V|n)}}$ is the frequency of v with respect to n, $\mathbf{idf_{(n|V)}}$ is the number of n with respect to v, and $\mathbf{AD_{(V|n)}}$ is the average distance between v and n.

3.4.1.3 The Weighting Process of OWS

The OWS weighting process is designed in Figure 3.5:

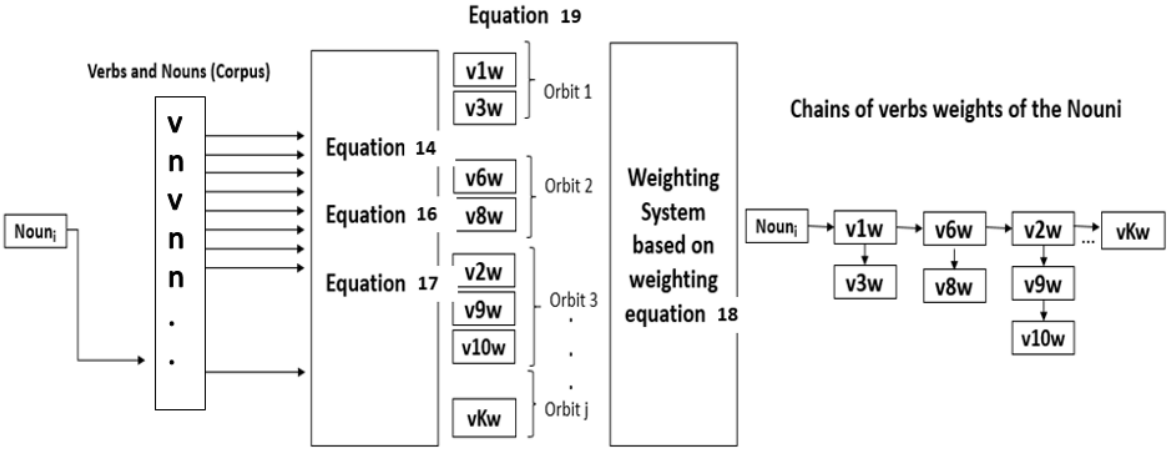


Figure 3.5 OWS Process Architecture

In Figure 3.5, the list of verbs that appeared with the noun Nouni are weighted based on equations 14, 16, 17, and 18; then, the weights are distributed to the appropriate orbits based on equation 19.

In Figure 3.5, orbit 1 represents the inner orbit that contains the verbs that gained the highest weights, and orbit j represents the outer orbit that includes the smallest weights. The OWS generates the output as a linked list, and

the lists reflect the orbit's representation of the noun-verbs combinations. Nested linked lists are built for the noun being processed, and the list contains the verbs associated with the noun in the corpus.

Each level in the list represents one orbit, and all the verbs found in one orbit have roughly the same weight value (located with the range specified in equation 19). The verbs closed to the root node (noun node) have the largest weights, and the weights start to decrease toward the leaf.

The weighting process is designed as follows:

1. Compute the weighting parameters ($\mathbf{fr}_{(\mathbf{v}|\mathbf{n})}, \mathbf{idf}_{(\mathbf{n}|\mathbf{v})}, \mathbf{and}\ \mathbf{AD}_{(\mathbf{v}|\mathbf{n})}$) for each v appeared with n.
2. Compute the weight using equation 18 for each v appeared with n.
3. Construct the vector of n.
4. Specify the range of weights that should be included in each orbit. All the weights will be located between the interval (MIN_w, MAX_w), where MAX_w is the weight of the verb appears in the inner orbit, and the MIN_w is the weight of the verb appears in the outer orbit. Therefore, we assumed that the range would be:

$$\boldsymbol{Layer = Range\ of\ values\ in\ each\ orbit}$$

$$\boldsymbol{Layer = \frac{MAXw - MINw}{number\ of\ verbs}} \ldots (\mathbf{19})$$

(1) Extract all the nouns that share the verbs that appeared in the first three layers specified in step 4. Here it is important to mention that the NBDV method processes the nouns found in the first, second, and third layers, and three is set as the threshold value of the number of layers processed by the NBDV method. The identification of this threshold value is made by measuring the number of nouns processed based on the first five layers for 50 values of n (for 50 runs of the NBDV), and it was found that the number of nouns becomes very large after the third layer.

(2) Using OWS equation 18, compute the weights of each verb located in the first three layers with respect to each noun extracted in step number 5, and generate a vector of weights for each noun.

At the end of the OWS phase, the NBDV model has a number of vectors equals the number of nouns shared between the verbs located in the first three layers. The outputted vectors are then transferred to the second phase of the NBDV model.

3.4.2 Synonyms Detection Phase

The purpose of the second phase is to generate the required synonyms set. The input of the synonyms detection phase is a set of vectors generated from the OWS phase. The system now can investigate the vectors and compute the cosine similarity between them, as shown in Figure 3.6. The synonyms detection process can take the following steps:

(1) Find the similarity between $\vec{\mathbf{n}}$ and all the vectors $\vec{\mathbf{n}}_{\mathbf{x}}$ produced from the OWS phase using equation 20.

$$\boldsymbol{sim}(\vec{\boldsymbol{n}},\vec{\boldsymbol{n}}_{\boldsymbol{x}}) = \boldsymbol{cos}(\vec{\boldsymbol{n}},\vec{\boldsymbol{n}}_{\boldsymbol{x}}) = \frac{\vec{\boldsymbol{n}}.\vec{\boldsymbol{n}}_{\boldsymbol{x}}}{|\vec{\boldsymbol{n}}|.|\vec{\boldsymbol{n}_{\boldsymbol{x}}}|}$$

$$\boldsymbol{sim}(\vec{\boldsymbol{n}},\vec{\boldsymbol{n}}_{\boldsymbol{x}}) = \frac{\sum_{i=1}^{j} \boldsymbol{w}(\boldsymbol{nv_i}).\boldsymbol{w}(\boldsymbol{n_x v_i})}{\sqrt{\sum_{i=1}^{j} \boldsymbol{w}(\boldsymbol{nv_i})^2}.\sqrt{\sum_{i=1}^{j} \boldsymbol{w}(\boldsymbol{n_x v_i})^2}} \ldots (\mathbf{20})$$

Where j is the number shared verbs identified in OWS phase, $\mathbf{w}(\mathbf{nv_i})$ is the weight of the verb v_i with respect to the noun n, and $\mathbf{w}(\mathbf{n_x v_i})$ is the weight of the verb v_i with respect to the noun n_x. The sim (n, n_x) = 1, if all the verbs are shared and have the same weights. (0 <= sim (n , ni) <=1).

(1) Sort the similarity values in descending order. Discard all the similarity values that are less than 18%, this value was determined by scanning the value of the first 100 nouns.

(2) Consider the top values and extract the nouns corresponding to them as synonyms.

Back to orbit representation, the computation function takes the orbit number into consideration because the verbs in the inner orbits should have heavier weights. The equation solves the problem that may arise if a large number of verbs were shared at the outer orbits because their values will be very small to make any difference. This feature distinguishes our equation from the similarity equations used in NLP and IR fields. All the similarity equations such as the Dice's Coefficient, Jaccard's Coefficient, and the Cosine similarity deal with a set of values, not an ordered list, they do not consider the position of the values in the set before computing the similarity.

In steps two and three, the similarity values are sorted, and the top n values are treated. The value of n can be user-defined, but in the experiment chapter, we chose seven as the value of n (maximum seven synonyms for each noun). Each similarity value is a value that measures the closeness between two nouns. If the value is large, the NBDV method takes the participating nouns as synonyms.

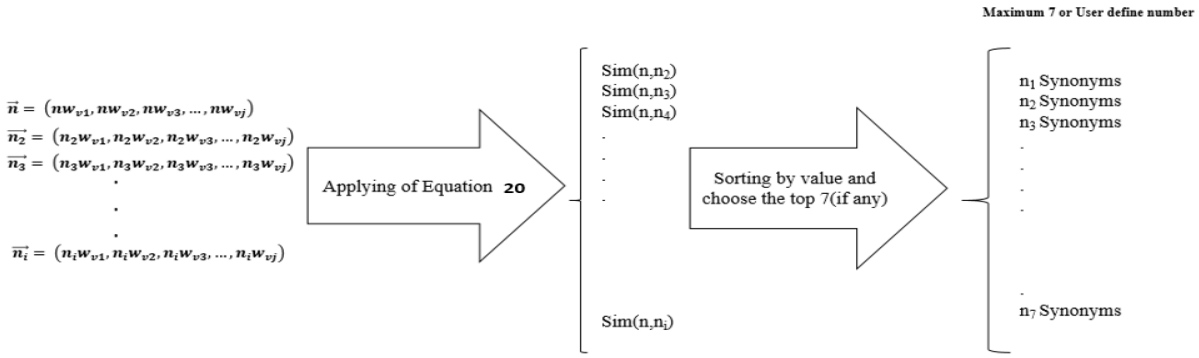


Figure 3.6 Synonyms Detection Steps Embedded in the NBDV Model

The NBDV method is completely statistical and language-independent for the languages that have the following word orders: SVO, VSO, VOS, OVS, but it is not applicable to the languages that have the SOV (like Hindi, Japanese, and Korean) and OSV (like Warao) word orders, because the OWS assumes that the verb precedes the noun.

3.4.3 NBDV Algorithm

The algorithm appears in Figure 2.7 is designed based on the methodology described in sections 3.4.1 and 3.4.2. The algorithm accepts two inputs, the noun being processed (called x), and a preprocessed huge corpus (called K). The preprocessing of the corpus explained in the experiment setting section 4.2. The algorithm creates two dynamic arrays, verbs(x) stores the verbs adjacent to x, and candidate(x) stores the candidate synonyms of the noun x. The output that represents the synonyms for the noun x (maximum seven elements) is retrieved in a static array S. S is an ordered list sorted by the cosine similarity (equation 20) between the noun x and the candidate synonyms x_c in S. In the NBDV algorithm, the OWS process is used twice, the first use is to compute the weights of the verbs adjacent to the main noun x (stored in verbs(x)), and the second use is for each noun x_c stored in candidate(x). The second use of the OWS will not hurt the time complexity because the algorithm performs it only for the verbs stored in verbs(x) after deleting all the verbs that are not located in orbit 1 -3 (called O_1, O_2, O_3).

The NBDV method of synonyms extraction is implemented in the VSyn software package (see Figure 4.1 in section 4.2.2). The purpose of developing VSyn software is to test the performance of the NBDV.

The VSyn is designed and implemented according to the detailed specifications of the NBDV algorithm described in section 3.4. The VSyn allows the user to enter the noun and search for the synonyms.

The phases of the NBDV methods were developed based on the methodology described in section 3.4 and implemented using VB 2013 programming language.

Algorithm: NBDV Extraction

Input: the noun x , preprocessed Text Corpus K // Kalimat corpus with term-tag-stem format
Output: the set S of synonyms of the noun x of the form: S = {syn_1, syn_2 ,…, syn_j}, where sim(x,syn_1) <= sim(x ,syn_j) <= sim (x ,syn_i) , with max (i) = 7 , j = 1 , 2 , … , max (i)
Process: applying the OWS weighting scheme, and finding the similarity based on equation 20.

Method:
Begin
Construct vec(x)={}; // vector of x contains the OWS weights of the verbs adjacent to x
Construct verbs(x)={}; // creating dynamic array to store the verbs adjacent to x
Search through K;
c = 0 ;
For each v_i appeared within 5 locations of x
{
Add vi to verbs(x)
Find fr(vi|x) // using equation 14
Find idf(x|vi) // using equation 16
Find AD(vi|x) // using equation 17
Find W(vi|x) // using equation 18
Add W(vi|x) to vec(x); vc = vc + 1; // vc : verbs counter
}
Layer = (max(w) – min(w)) / vc ; // computing the orbit range
For each w ∈ $\boldsymbol{vec}(\boldsymbol{x})$ do
{
if max(w)>= w >= max(w) – layer add w to O_1; // inner orbit
*Else if max(w)-layer> w >= max(w) – 2*layer add w to O_2;*
*Else if max(w)-2*layer> w >= max(w) – 3*layer add w to O_3 ; // the outer orbit*
Else delete w from vec(x); delete v from verbs(x); //excluded all the verbs that located after orbit 3
}
Construct candidate(x) // creating dynamic array to store the candidate synonyms of x
For each vi ∈ $\boldsymbol{verbs}$ ($\boldsymbol{x}$) do
If noun x_c adjacent to vi add xc to candidate(x) // add noun x_c the candidate array
For each x_c ∈ $\boldsymbol{candidate}$($\boldsymbol{x}$) do
{
compute the weights of each vi $\boldsymbol{verbs}$ ($\boldsymbol{x}$) with respect to x_c
construct vec(n_c);
if sim(vec(x) , vec(x_c)) < 0.18 delete x_c from the candidate(x) // similarity equation 20
}
Sort candidate (x);
S = the first seven element of candidate(x); // the final synonyms set
End

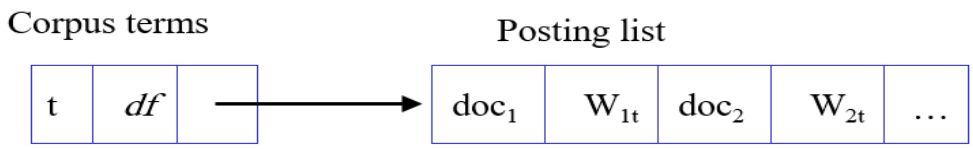
Figure 2.7 The NBDV Algorithm for Synonyms Extraction

3.5 IR System Based in VSM Model

As mentioned in the introduction section of this chapter, the IR system used in our method of retrieval is developed based on the VSM model. Section 3.2 explains the reason behind the selection of this model. This section gives a detailed description of the VSM model from the IR point of view (because section 3.3.1 described the use of the VSM model as a text extraction model, and we see how equation 7 which represent that cosine similarly between two sentences is used to measure the resemblance between two sentences). The generalization of equation 7 leads to measure the similarity between any two pieces of text (two documents, two sentences, document and sentence, user query and document). The flexibility of the VSM model allows the use of the VSM in all the fields of text mining, and all we need is to redefine the weighting parameters, the weighting scheme, and the similarity equation to reflect the specialty of the text mining application being developed.

In our method, the VSM is returned to its origin since the development of the VSM was first proposed in the field of IR (Salton, Wong, & Chungshu, A vector space model for automatic indexing, 1975). The theoretical background of the VSM model in IR is described in section 1.3.3. In our work, and the actual steps in designing the IR system came as follow:

Inverted Index Creation: we built several inverted indexes, and the reason for that is explained in the results and discussion chapter, but all of them are constructed using the same weighting scheme and the same structure, see the next form



Where t represents the term found in the corpus of original documents (Document-based index in Figure 3.1) or the term found in the corpus generated from the MLS extractor (Extract-based index in Figure 3.1), the df is the document frequency of the terms t or the number of documents that contains t, and the W_{it} is the weight of term t in document i, and we used the best-known weighting scheme proposed by Salton in (Salton, Wong, & Chungshu, A vector space model for automatic indexing, 1975), it called the tf.idf weighting scheme, where tf is the frequency of term i and df is the number of documents that contain i,

$$W_{it} = (1 + log\, f_{it})\, log \frac{N}{df} \ldots (21)$$

Where w_{it} is the weight of the term t in text segment i, f_{it} is the frequency of the term t in text segment i, df is the number of text segments contain t, N is the number of text segments in the corpus; text segment could be query, document, or summary in case of extract-based index.

Similarity Computation: After computing the weights and creating the inverted indexes, we calculated the similarity between each document and the user query by computing the cosine of the angle between the vectors that represent them (Schütze, Christopher, & Prabhakar, Introduction to information retrieval, 2008) .

$$sim(\vec{d_j}, \vec{Q}) = cos(\vec{d_j}, \vec{Q}) = \frac{\vec{d_j}.\vec{Q}}{|\vec{d_j}|.|\vec{Q}|} = \frac{\sum_{i=1}^{n} wd_{j_i}.wQ_i}{\sqrt{\sum_{i=1}^{n} wd_{j_i}^{\,2}} \cdot \sqrt{\sum_{i=1}^{n} wQ_i^{\,2}}} \ldots (22)$$

Where $\vec{\mathbf{d}}_\mathbf{j}$ is the vector of document j, $\vec{\mathbf{Q}}$ is the vector text query Q, $\mathbf{wd_{j_i}}$ is the weight of the term i in $\mathbf{d_j}$, $\mathbf{wQ_i}$ is the weight of the term i in Q, n is the number of terms in the whole corpus

Ranking the retrieved documents: the system returns the documents that match the query based on the similarity computed in step 2. The retrieved set of documents is sorted in descending order based on the similarity values.

Query Expansion: the expansion of the user query is performed by appending the user query terms with the first and second synonyms generated from the NBDV method like the following example:

The original query	الحاسوب computer		مواد materials	تدريس Teaching	
The expanded query	معالج processor	كمبيوتر computer	No synonyms generated	عمل work	تعليم educate

3.6 MLS and NBDV Merits and Deficiencies

This chapter described the methods that are used to boost the IR system. These methods are the MLS text extraction method and the NBDV synonyms extraction method. For these methods, we can conclude the following merits:

The semantic investigation of the text contents: the semantic investigation in our methods of text extraction comes in the construction of the generic summaries and in the extraction of query synonyms. The MLS extraction differs

from similar work in the field in using the latent semantic analysis in the construction of the automatic extracts. This feature cannot be found in (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001), and (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013) (these references are mentioned in the introduction as the works used the generic summaries in reducing the inverted index size). Also, the NBDV synonyms extraction method used the distinctive verbs to find the semantic relations to link the query terms with semantically related nouns.

The efficiency constraints considered in the MLS and NBDV methods. Note that the design of the MLS recognizes the high time complexity of the latent semantic analysis. So we design a framework of semantic analysis that uses the LSA only for complicated cases that the traditional statistical techniques (Jaccard Coefficient and the VSM) cannot deal with them. Also, the NBDV involved the use of a new weighting scheme (OWS) that substituted the traditional tf.idf weighting scheme used in the CBoW and SG models ((Mikolov, Chen, Corrado, & Dean, 2013), (Leeuwenberga, Vela, Dehdar, & Genabith, 2016)). The tf.idf weighting scheme in ASE is time-consuming and takes $O(n^2)$, as we will see in the evaluation section, and our OWS takes O(n).

Drawing a map for using the statistical methods in text mining in a practical way. Our method of text extraction process the text in a hierarchal statistical structure. The lower layer in the MLS method processes the text based on the verbatim similarity, and the middle layer processes the text based on the importance of the term for the segment of text, and the upper layer processes the text based on the semantic meanings of the concepts or topics. In this way, we handle all the text segments based on the verbatim, terms importance, and semantic features in the text. The hierarchal statistical framework assures that we can benefit from the advantages of different statistical techniques.

However, the MLS and the NBDV have the following limitations:

MLS does not solve the polysemy problem. The polysemy problem arises when one word has more than one meaning, as we said in the introduction chapter. This problem can be solved by the latent semantic analysis because the latent semantic analysis processes the words relative to their context. But, in the MLS extraction such kind of words are processed in the lower or middle layers, and it will not reach the LSA layer (e.g if two sentences have the word „bank“ in the same document but with two different meanings then the MLS will process them in the first

or second layer because the verbatim situation exists and the term frequency and term distribution will be reasonable).

NBDV does not solve the synonyms of verbs and adjectives. In the design of the NBDV for synonyms extraction, we used the verbs to find the synonyms of the nouns, but if the user query contains verbs or adjectives, no expansion will happen.

NBDV can not process the word orders that end with the verbs (OSV, SOV). The OWS scheme upon which the NBDV method is built assumes that the verbs precede the nouns (Subject, Object). This drawback prohibited the use of the NBDV in certain languages such as Hindi, Japanese, Korean.

The next chapter will explain the experiments used to test the MLS and NBDV models as stand-alone systems and as tools to boost the IR system. Also, the next chapter will present the results that were collected from each experiment for further evaluation and analysis.

CHAPTER 4 EXPERIMENTS and RESULTS

In this research, the MLS method developed in chapter three is used to summarize the documents before used them to construct the inverted index. And, the NBDV method of synonyms extraction is used to expand the user query. The output of the MLS and NBDV are used to support the relevancy and efficiency of the IR system designed in section 3.5 of our methodology chapter. Both of the MLS and NBDV perform the extraction semantically and efficiently, and the results obtained in the conducted experiments prove that the relevancy measurements such as the recall and precision and the efficiency rates are reasonable and beneficial.

This chapter explains the series of experiments used to test the models that are proposed and developed in chapter 3. In section 4.2, we listed the datasets and the experiments' environment. And, in section 4.3, we present the results that were collected from each experiment for further evaluation and analysis.

4.1 Introduction

The testing of the effectiveness of the MLS and NBDV extraction methods on the relevancy and efficiency of the IR system demands to conduct a series of experiments and to collect the results for further analysis. The general headlines of the experiments can be summaries in the following three points:

MLS experiment: The purpose of the experiment is to test the accuracy of the MLS extractor using intrinsic approaches. The relevancy measurements (recall, precision, and condensation rate) results are collected for evaluation; the relevancy measures are computed based on the gold summaries prepared manually in well-known Arabic datasets.

NBDV experiment: The purpose is to test the NBDV method using intrinsic evaluation. We manually and automatically collected the precision and recall based on a comparison of the synonyms found in well-known Arabic language dictionaries.

IR experiments: A series of experiments with different conditions and constraints have been held. We collected the results of the relevancy measurements and the size of the inverted index of the IR system before and after applying the MLS and NBDV methods.

The experiments are designed to assess the developed methods and to satisfy the objectives that are proposed at the beginning of the research. For example, the MLS and NBDV experiments are designed to check the satisfaction of the objectives numbers 3 (“Building an effective text summarizer using the efficient framework of semantic analysis”) and 7 (Developing an efficient synonyms extraction model, and employ this model in a synonyms extraction system that extracts synonyms for the user query terms) because we used the intrinsic approaches that can test if MLS and NBDV method obtained accurate extraction or not. Also, for the objective number 4 (“Proving that the use of the traditional statistical bag of word models (such as theVSM and Jaccard coefficient) are not suitable for performing reasonable text summarization especially to reduce the inverted index in an IR system“), we implemented the MLS extractor with three other extractors that are based on the Jaccard coefficient, VSM, and LSA in separate manner, and we used the same experiment setting and environment to test the four extractors, and we collected the results for intrinsic and extrinsic evaluation. The IR experiment is designed to assess the objective 5 (Improving the retrieval time through the reduction of the index size which will be constructed from the summaries instead of the original documents) , 6 (Analyzing the efficiency of Information Retrieval systems with and without Automatic Text Summarization using IR evaluation measures) , 8(enhancing the user query with the synonyms generated automatically and test their relevancy on the IR system that uses the summaries as a source of the index), and 9 (Estimating the effectiveness of our summarizers using extrinsic methods by evaluating their influence on Arabic information retrieval performance) because the experiment integrated the MLS and NBDV methods developed in this research with the IR system designed in section 3.5, and the experiment is conducted with and without MLS extraction, and with and without synonyms expansion, and in all the cases, we collected the required results for evaluation.

This chapter is organized as follows: section 4.2 depicts the datasets and the environment of the experiment. Section 4.3 describes the MLS experiments and the collected results. Section 4.4 shows the NBDV experiment and the collected results, and section 4.5 explains the IR experiments and the obtained results.

4.2 Experiments Environment

The Arabic Language has been chosen as the language of the case study, because one of the primary objectives of this research is to measure the effect of the MLS model and NBDV model on the relevancy of the Arabic IR systems that use the Vector Space model, and to measure the accuracy of applying the MLS model on the recall

and precision of the Arabic language text extraction systems. But, to diversify the test conditions and environment, we chose to test the developed text extraction methods over another language, and the English language has been chosen because of its spread over the world.

4.2.1 Datasets

At first, the experiments to test the effectiveness of our method was applied to four datasets for the Arabic and English Languages.

1. Essex Arabic Summaries Corpus: This Corpus is published free on http://www.lancaster.ac.uk/staff/elhaj/corpora.htm. The Corpus contains 153 Arabic articles and 765 human-generated extractive summaries. For each document, there are five manual extracts. The corpus contains documents with different subject areas, including art, music, science and technology, education, finance, health, politics, and religion. The Corpus used recently by Al-Radaideh and Bataineh in (Al-Radaideh & Bataineh, 2018).

2. Kalimat data corpus[11] : Kalimat contains 20,291 Arabic article (3,537,677 Noun, 1,845,505 Verb, 115225 adjectives, and totally 6,286,217 terms). The corpus comprises greater than 6,000,000 terms. The data was taken from Omani newspapers. We tried to vary the topics and the domain of knowledge, so the selected data talking about health, science, history, art, religion, technology, environment, economic, and financial aspects.

3. 242 data corpus: The corpus includes 242 Arabic text documents, 60 queries with their manual relevancy assessments. The corpus used by many researchers who investigated the Arabic IR field (Hanandeh, 2013).

4. The Blog Authorship corpus: It's an English language corpus that collects the posts of 19,320 bloggers and contains 681,288 text document composing 140 million words (Schler, Koppel, Argamon, & Pennebaker, 2006).

[11] The corpus is available free on http://www.lancaster.ac.uk/staff/elhaj/corpora.htm

4.2.2 Experiment setting

All the experiments were performed on Intel® Core™ i5-7200U CPU @ 2.5GHz processor with 8 GB RAM and Windows 10 OS. The MLS and the NBDV and the IR methods were developed based on the methodology described in chapter 3 and implemented using VB 2013 programming language.

4.2.2.1 MLS experiment setup

Regarding the MLSextractor, LSAExtractor, VSMExtractor, and JacExtractor, we used Visual Basic to implement them, with Excel sheets as an interface. To make the MLS experiment more reliable, we linked our software with the Latent Semantic Analysis Software developed by the University of Colorado Boulder. Also, we used Koja stemmer to produce the Arabic stems, and we used Porter Stemmer to produce the English stems. The evolution of the obtained results was conducted using the Containment Evaluation and ROUGE 2.0 evolution tool. As described in chapter 1, ROUGE is an evaluation tool for summarization tasks. It measures the quality of automatic summary by comparing it with reference summaries generated manually. In the Literature, ROUGE widely used to evaluate the summarization systems, and from our list of references, the ROUGE was used in (Svore, Vanderwende, & Burges, 2008), (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), (Ferreira, et al., 2013), (Wang & Ma, 2013), (Sankarasubramaniam, Ramanathan, & Ghosh, 2014), and (Ba-Alwi, Gaphari, & Al-Duqaimi, 2015). In our ROUGE runs, we evaluate each automatic summary with reference summaries founded in Essex and Kalimat, and the recall, precision, and F_measure values were collected. ROUGE 2.0 computes the recall and precision according to the following equations (Lin C. Y., 2004).

$$\boldsymbol{Recall} = \frac{\boldsymbol{number\ of\ overlapping\ unigrams}}{\boldsymbol{total\ number\ of\ unigram\ in\ reference\ summary}}$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{number\ of\ overlapping\ unigrams}}{\boldsymbol{total\ number\ of\ unigram\ in\ system\ \ summary}}$$

$$\boldsymbol{F-score} = \boldsymbol{2} * \frac{(\boldsymbol{Recall} * \boldsymbol{Precision})}{(\boldsymbol{Recall} + \boldsymbol{Precision})}$$

After implementing the extraction systems, we started our experiment procedure, and we followed the following steps:

Step1: Text preparation and preprocessing:

1. Stemming the text's words: Load each document to Khoja Stemmer software and generate a list of stems.
2. Stopwords removal: in this step, we removed all the stopwords such as "الى","على","عن","من".

3. Punctuation marks removal: such as “ً”,”ٌ”,”ٍ”,”ُ”.
4. Non-letter symbols removal: such as “(”,”$”,”+”,”#”.
5. Foreign words removal: because the datasets used in this experiment are taken from the Arabic language all the English text is considered foreign.
6. Text normalization: such as replacing :
    - letters “ ء , إ , أ “ by “ا".
    - letter “ة” by “ه”.
    - letter “ئ,ى ” by “ي”.
7. Terms’ frequencies and distributions calculations.

After applying the pre-processing stage, the generated sentences and terms are organized in excel sheets. The sheets will be the input for the four text extractor (JacExtractor, VSMExtractor, LSAExtractor, and MLSExtractor) because they contain the required parameters (doc id, term, stem, term frequency, and term distribution) and their values, which are necessary to initiate the second stage in each extractor.

Step2: similarity matrix building

The data in the sheets are entered into the four extractors to generated the sentences’ similarity values. Then, the similarity values are organized in two-dimensional matrixes. Table 4.1 shows the matrix template that was used to collect the similarities among the document sentences in the four extractors. The x's characters represent the similarities values. Where $\mathbf{0} <= \boldsymbol{x} <= \mathbf{1}$. Four different matrices from the four extraction algorithms - discussed in section 3- were generated for each document.

Table 4.1 Sentences' Similarity Matrix Template

S1	X	x	x	x	x	x
S2		x	x	x	x	x
S3			x	x	x	x
.				x	x	x
.					x	x
.						x
Sn	S2	S3	.	.	.	Sn

Step 3: Deleting similar sentences.

In this step, we initiated the deletion process. We investigated each value of x appeared in Table 4.1 and decided to delete or not the corresponding sentence depending on the conditions discussed in Section 3.3. The deletion process is

part of the four extractors that are developed in the research, and it works under the same conditions. Before preceding to the last step, we present a practical example. The example practices the deletion process in the VSMExtractor for document 17 from Essex corpus. Table 4.2 details the attributes of document 17. In this example, we present in detail the deletion steps of the redundant sentences in document 17, this document contains a moderate number of sentences (14) and words (410), and the document subject (Environment) is a general subject. However, we can replace document 17 with any other document.

Table 4.2 Essex Corpus - Document 17

Attribute	Value
Document Subject	Environment
Source	Essex Corpus
Title	Chemistry (الكيمياء)
Number of sentences	14
Number of words	406

The Document 17 sentences in both Arabic and English are listed below:

1. الكيمياء هى في الأصل كلمة عربية مثل السيمياء، مأخوذة من الكمي وهو الشجاع، و المتكمي في سلاحه أي المتغطي المتستر بالدرع والبيضة، وسُمِيت كذلك لأن الكيمياءيين القدماء كانوا يحتفظون بمعلوماتهم سرية عن الآخرين، وتعنى كمصطلح العلم الذى يدرس المادة وتفاعلاتها وعلاقاتها بالطاقة.

2. ونظرا لتعدد وإختلاف حالات المادة، والتى عادة ما تكون في شكل ذرات، فإن الكيميائين غالبا ما يقوموا بدراسة كيفية تفاعل الذرات لتكوين الجزيئات وكيفية تفاعل الجزيئات مع بعضها البعض.

3. والكيمياء هو علم يدرس العناصر الكيميائية والمواد الكيميائية و التركيب والخواص والبناء والتحولات المتبادلة فيما بينها اي التفاعلات الكيميائية.

4. حاول الإنسان عبر العصور أن يبحث في طبيعة العالم الذي حوله، وذلك بدافع غريزة حب المعرفة،ومن خلال ذلك، تم الكثير من الاكتشافات المهمة التي ساعدت على تطوير العلوم والتكنولوجيا ومن ضمنها علم الكيمياء وهو علم يعنى بطبيعة المادة ومكوناتها، وكذلك بكيفية تفاعل المواد المختلفة مع بعضها بعضاً، وعلى هذا تكون وظيفة العالم الكيميائي الأساسية هي معرفة أكبر قدر ممكن من المعلومات عن طبيعة المادة التي أوجدها الله في هذا الكون.

5. بدايات علم الكيمياء.

6. تعود بدايات علم الكيمياء إلى زمن موغل في القدم، فلقد أختلف في مكان نشأته، قيل أن بداياته كانت في القرن الثالث قبل الميلاد، كان دبغ الجلود وصناعة الأصباغ ومستحضرات التجميل من بين الفنون التي مارسها المصريون.

7. مساهمة العرب في تطوير الكيمياء.

8. عندما فتح العرب مصر سنة ولا ريب أن أولئك الفاتحين أسهموا بقدرٍ موفور في تطوير الكيمياء، حيث يعتبرون أول من اشتغل بالكيمياء كعلم له قواعده وقوانينه، وذلك منذ القرن الثاني الهجري، وطبقوا إنتاجهم في الصيدلية بصفة خاصة .

9. وما زال الإلتحام بين شتى المفاهيم لعلوم الكيمياء القديمة ينم عن اللفظ العربي نفسه مثل ألوخيمياء .

10. كذلك أصل كلمة كحول وهو عربي بمعنى غول وغرّبت هذه الكلمة أو حولت على اللغة الغربية بهذه الصفة.

11. و استمرت أصول الكيمياء العربية مرجعاً للغرب إبان القرون الوسطى وانتقلت ترجمات أعمالهم إلى أوروبا في القرن الثاني عشر الميلادي والتي اشتهرت بعد أن وصل الفتح العربي إلى الأندلس سنة711م يحمل معه المعارف العربية.

12. وفي الجامعات العربية ببرشلونة وطليطلة تعلم طالبوا العلم من جميع أنحاء أوروبا فن الكيمياء.

13. الكيمياء الحديثة.

14. يرجع تاريخ الكيمياء الحديثة إلى القرن السابع عشر الميلادي بأبحاث بويل الذي قسم الأجسام إلى مواد أولية عناصر ومركبات و مخاليط و تلت أبحاث بلاك، ولافوازيية عن الاحتراق والتأكسد ثم برتلي الذي اكتشف الأكسجين في الهواء ، ثم كافندش الذي اكتشف تكوين الماء ثم دالتون الذي وضع النظرية الذرية عن تكون المادة وتعرّف الكيمياء الحديثة بأنها علم طبيعي في تكوين المادة.

1. Chemistry is originally an Arabic, which was taken from the quantum, which means brave who concealed his weapon and covered by a shield. And, it was also named because the ancient chemists kept their information confidential, and the term Chemistry means the science that studies the matter and its interactions.

2. Because of the multiple states of matter, usually in the form of atoms, chemists often study how atoms interact to form molecules and how molecules interact with each other.

3. Chemistry is a science that studies chemical elements, chemical composition, properties, construction, mutations, chemical interactions.

4. Throughout the ages, man has tried to look at the nature of the world around him, motivated by the instinct of love of knowledge. The basic function of the chemical world is to know as much information as possible about the nature of matter created by God in the universe.

5. The beginning of chemistry.

6. The beginnings of chemistry date back to a long time ago. With a mysterious place of origin, it was said that its beginnings in the third century BC. Leather tanning, dyes, and cosmetics were among the arts practiced by the Egyptians. The skill of the Egyptians was subdued with the theories of the Greeks, which led to the emergence of those practicing chemistry.

7. The contribution of Arabs to the development of chemistry.

8. When the Arabs conquered Egypt, there is no doubt that these conquerors contributed significantly to the development of chemistry, where they are considered the first to work in this science with its rules and laws. Since the second century, they applied their production in Pharmacology in particular.

9. The various concepts of ancient chemistry still use Arabic terms such as alchemy.

10. The origin of the word alcohol is Arabic, and it comes from the word Gul, and this word was transferred to the Western language.

11. Arab chemistry continued to be a reference to the West during the middle ages.

12. At the Arab universities in Barcelona and Toledo, science students from all over Europe learned the art of chemistry.

13. Modern Chemistry.

14. Modern chemistry dates back to the 17th century by Boyle's research, which divided objects into raw materials, compounds, mixtures, and followed Black's non-invasive research on combustion and oxidation, then Bartley, who discovered oxygen in the air, then Cavendish, who discovered the formation of water and then Dalton, who developed Atomic theory about the formation of matter. Modern chemistry is defined as a natural science that studies the composition of matter.

After computing the necessary parameters, we can apply equations 3 and 7 through the first and second stages in our algorithms and fill the similarity matrix, as shown in Table 4.3. The deletion process uses the similarity values that are generated in step 2 and omits the sentences that have high similarity with the existing sentence. Table 4.3 presents the VSM similarity matrix of the sentences of document 17 (the sentences' similarity were generated in the VSMExtractor). The intersection cells' values represent the cosine similarity values. For example, the similarity between sentence 1 and sentence 3 is 88%, and between sentence 7 and sentence 13 is 20% and so on.

Table 4.3 the VSM similarity table for Document 17 – Essex Corpus

	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
S1	43%	→88%	43%	→80%	26%	29%	6%	→98%	44%	7%	26%	→54%	26%
S2		→76%	→75%	0%	13%	0%	0%	7%	0%	0%	0%	0%	45%
S3			→77%	→99%	33%	5%	42%	31%	0%	1%	32%	7%	→100%
S4				→70%	26%	20%	19%	15%	6%	9%	16%	3%	62%
S5					→78%	20%	4%	→98%	0%	5%	→100%	26%	46%
S6						1%	27%	→77%	7%	22%	8%	2%	32%
S7							→83%	43%	44%	37%	44%	20%	2%
S8								27%	18%	40%	10%	5%	25%
S9									12%	10%	38%	6%	10%
S10										31%	13%	0%	0%
S11											18%	5%	29%
S12												6%	10%
S13													→98%

The table shows that the similarity values between the pair of sentences (1, 3), (1, 5), (1, 9), and (1, 13) exceeds the threshold value (50%), which means that we can delete the sentences 3, 5, 9, and 13. As shown in Table 4.4.

Table 4.4 The VSM similarity table for Document 17 – after deleting 3, 5, 9, 13

	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
S1	43%	→88%	43%	→80%	26%	29%	6%	→98%	44%	7%	26%	→54%	26%
S2		→76%	→75%	0%	13%	0%	0%	7%	0%	0%	0%	0%	45%
S4				→70%	26%	20%	19%	15%	6%	9%	16%	3%	62%
S6						1%	27%	→77%	7%	22%	8%	2%	32%
S7							→83%	43%	44%	37%	44%	20%	2%
S8								27%	18%	40%	10%	5%	25%
S10										31%	13%	0%	0%
S11											18%	5%	29%
S12												6%	10%

The sentences (2, 3) and (2, 4) have significant similarities, but 3 was discarded in the previous step so we can remove only 4 from the similarity table. As shown in Table 4.5.

Table 4.5 The VSM similarity table for Document 17 – after deleting 4

	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
S1	43%	→88%	43%	→80%	26%	29%	6%	→98%	44%	7%	26%	→54%	26%
S2		→76%	→75%	0%	13%	0%	0%	7%	0%	0%	0%	0%	45%
S6						1%	27%	→77%	7%	22%	8%	2%	32%
S7							→83%	43%	44%	37%	44%	20%	2%
S8								27%	18%	40%	10%	5%	25%
S10										31%	13%	0%	0%
S11											18%	5%	29%
S12												6%	10%

Finally, the pair (6, 9) has a 77% similarity value, but 9 was discarded before. The pair (7, 8) has 83% similarity value and sentence 8 will be deleted. The pair (13, 14) has a 98% similarity value, but we cannot delete sentence 14 because the base sentences 13 is deleted in the first iteration of the deletion process. The remaining sentences

which will st1, st2, st6, st7, st10, st11, st12, and st14, and these sentences will construct the VSM automatic extract. See Table 4.6.

Table 4.6 The Similarity Matrix of Document 17 after Completing the Deletion Process.

	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
S1	43%	88%	43%	80%	26%	29%	6%	98%	44%	7%	26%	54%	26%
S2		76%	75%	0%	13%	0%	0%	7%	0%	0%	0%	0%	45%
S6						1%	27%	77%	7%	22%	8%	2%	32%
S7							83%	43%	44%	37%	44%	20%	2%
S10										31%	13%	0%	0%
S11											18%	5%	29%
S12												6%	10%

Step 4: Extract Generation.

The remaining sentences after applying the deletion process are concatenated sequentially according to their orders in the original document. The resulted sentences represent the final summary. Note that four summaries will be generated for each document, one from each extractor.

4.2.2.2 NBDV experiment setup

The NBDV method uses unsupervised learning to extract nouns synonyms. All the machine learning approaches require huge data corpus, so Kalimat data corpus is used in the experiment of the NBDV. Kalimat contains 20,291 Arabic article (3,537,677 Noun, 1,845,505 Verb, 115225 adjectives, and totally 6,286,217 terms.

Regarding the NBDV synonyms extraction method, the phases of the NBDV methods were developed based on the methodology described in section 3.4 and implemented using VB 2013 programming language. The NBDV method of synonyms extraction with what they include of parameters and equations is implemented in the VSyn software package (see Figure 4.1). The purpose of developing the VSyn software is to test the performance of the NBDV method of synonyms extractions, so the VSyn is designed and implemented according to the detailed specifications of the NBDV algorithm described in section 3.4.3. The VSyn allows the user to enter the noun and search for the synonyms. The software interface includes three output panels, the first one displays the sentences that contain the noun, the second panel displays the verbs appeared with this noun sorted by their distance from the noun, and the third panel displays the generated synonyms that are sorted in descending order according to their similarity to the noun being processed. Also, the software generates an excel sheet; this sheet contains all the

processed verbs and their obtained weights. For automatic evaluation with automatic evaluation tools such as the ROUGE tool, the system generates a text file for each noun that contains its synonyms list.

In Kalimat, the terms are already tagged, but the preprocessing operations were performed to eliminate the Stopwords, punctuations markers, special and strange symbols. Also, a simple modification is made to unify all the subtypes under one tag. For example, Kalimat classifies the types of the nouns, for example اسم الآلة(the nouns that refer to equipment's or tools such as key, saw, lathe, fan, radiator, and scalpel) , اسما الزمان و المكان (nouns indicate the place and time of the action such as park, airport, appointment), and all the nouns types were unified under the "noun" tag. The same thing was done for the verbs; all verb types were unified under the tag "verb". After the preprocessing, the text is stored in (term-tag-stem) format. 564 nouns from Kalimat dataset randomly selected to extract synonyms for them. The nouns are processed one by one, and the generated synonyms were collected for evaluation. Also, the verbs and nouns processed in each run were collected to measure the processed portion of the corpus and to evaluate the time complexity needed to finish each single synonyms extraction operation. The types of results collected in the experiment with the purpose of each type appear in the next section.

After collecting the complete results, the results were evaluated in two separate ways. Firstly, by comparing our results with two online sources of Arabic Language synonyms; Almaany[12] and Google Translate. A sample of nouns found in our corpus was randomly chosen, and a comparison between the automatic synonyms sets generated by the VSyn system and the synonyms sets available on those two online sources has been established. Secondly, six Arabic language experts were hired to measure their degree of satisfaction with the accuracy of the VSyn system.

For accuracy comparisons with the other publications in the field of statistical synonyms extraction, the same relevancy measures are used to test the accuracy of the CBoW and SG model; the recall and precision.

The recall is the number of correct synonyms of a noun returned by our method relative to the number of actual synonyms number found in the Arabic language for that noun (taken from a base dictionary or an expert knowledge).

[12] https://www.almaany.com/ar/thes/ar-ar/

$$Recall = \frac{number\ of\ correct\ synonyms\ of\ the\ noun\ generated\ automatically}{actual\ number\ of\ synonyms\ of\ that\ noun\ found\ in\ the\ Arabic\ Language}$$

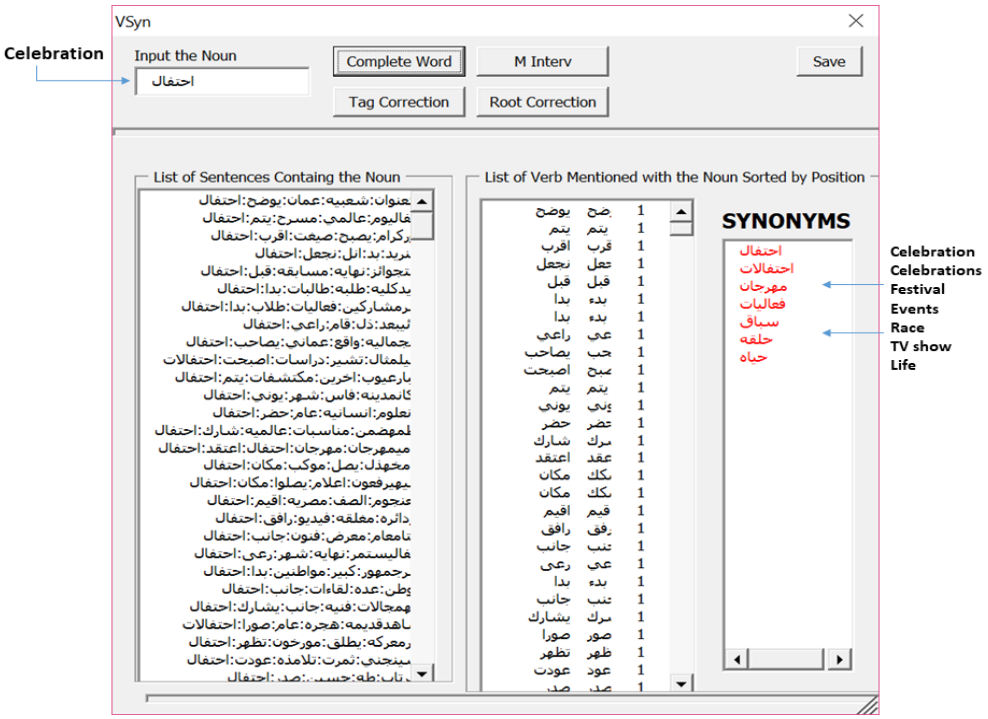


Figure 4.1 VSyn Interface

The precision is the number of correct retrieved synonyms of a noun relative to the total number of synonyms generated by the automatic synonym extraction system.

$$Precision = \frac{number\ of\ correct\ synonyms\ of\ the\ noun\ generated\ automatically}{number\ of\ synonyms\ of\ the\ noun\ generated\ automatically}$$

Also, to measure the behavior during the synonyms extraction process, and to assess the relevancy at each point, a new synonym is extracted, two new evaluation schemes are proposed:

- Average recall after the processing of the noun number *i*:

$$A(R|i) = \frac{\sum_{n=1}^{i} R_n}{i}$$

- Average precision after the processing of the noun number *i*:

$$A(P|i) = \frac{\sum_{n=1}^{i} P_n}{i}$$

The $A(R|i)$ and $A(P|i)$ trace the recall and precision trend during the processing of the nouns by the NBDV method.

The recall and precision are manually and automatically collected, and the automatic evaluation is performed using the ROUGE 2.0 tool and to measure the precision and recall using the ROUGE tool, the text file generated from the VSyn software is matched against the reference extract taken from the base dictionary.

4.2.2.3 The IR system experiment setup

Before proceeding, the following concepts have the following meanings in the explanation of the IR experiments.

- MC-based retrieval: the retrieval process that uses the main corpus to construct the inverted index (without summarization).
- MLS-based retrieval: the retrieval process that uses the summaries generated from the MLSextrcator to construct the inverted index.
- LSA-based retrieval: the retrieval process that uses the summaries generated from the LSAextrcator to construct the inverted index.
- VSM-based retrieval: the retrieval process that uses the summaries generated from the VSMextrcator to construct the inverted index.
- JAC-based retrieval: the retrieval process that uses the summaries generated from the JACextrcator to construct the inverted index.
- The MLS-based retrieval, LSA-based retrieval, VSM-based retrieval, and JAC-based retrieval are called extracts based retrieval.

The IR system described in section 3.5 is implemented using VB2013 with excel sheets as interfaces. Table 4.7 presents the VB developed functions and the role of each one.

Table 4.7 VB Functions Created in the IR System

VB function	Role
ExtractCorpusCreator	Constructs the extracts' corpus from the main corpus based on the sentences extracted from one of the extractors described in section 3.3.2 (MLSExtractor, LSAExtractor, VSMExtractor, and JacExtractor)
InvertedIndexCreator	Creates the inverted index of the main corpus
ExtractInvertedIndexCreator	Creates the inverted index from the extracts corpus constructed by the ExtractCorpusCreator
DocumentLength	Finds the length of each document (necessary for document length normalization)
TermWeight	Finds the weights of the terms based on equation 21 section 3.5
DocLenNormalize	Normalizes the term weight based on the output of the DocumentLength
Ir	Finds the similarity between the query and all the documents in the main or extract corpus-based in equation 22 section 3.5
RecallandPrecision	finds the recall and precision after the Ir running based on the definition of the recall and precision appeared in table 1.2 section 1.1.5
Inperppr	finds the Interpolated Average Precision after the RecallandPrecision running based on the definition of the Interpolated Average Precision appeared in table 1.2 section 1.1.5, and draws the precision-recall curve
MAP	finds the main average precision after the RecallandPrecision running based on the definition of the recall and precision appeared in table 1.2 section 1.1.5

The IR system experiments include the following experiments:

1. Exp1:
    - Purpose: compare the relevancy results obtained from the MC-based retrieval with the relevancy results obtained from the extract-based retrieval when the relevant documents are selected manually for each query.
    - The number of inverted indexes created is 5, one from the main corpus, one from the MLSextractor extracts, one from the LSAextractor extracts, one from VSMextractor extracts, and one from the Jacextractor extracts.
    - The number of processed queries is 60. the length of the queries distributed between 2 words such as „ تقنية المعلومات Information Technology“ to five words such as „ تمييز الاشكال بواسطة الحاسب الالي shapes recognition by the computer”. The queries are mentioned in the 242 corpus.
    - Manual relevancy assessment, for example, for the query “تقنية المعلومات” the following documents are recognized as relevant: 10, 96, 145, 175, 239. The relevancy assessment

of the queries was mentioned in the 242 corpus and done manually by the corpus developers.

- Arabic language datasets
- Without synonyms expansion

2. Exp2:
    - Purpose: measure how close the relevancy results for each extracts based retrieval to the relevancy results obtained from the MC-based retrieval (in simple words, measuring the effect of each extractor on the relevancy of the IR system)
    - The number of inverted indexes created is 5, one from the main corpus, one from the MLSextractor extracts, one from the LSAextractor extracts, one from VSMextractor extracts, and one from the Jacextractor extracts.
    - The number of processed queries is 100, and they were selected manually.
    - Automatic relevancy assessment, the retrieval list of the MC-based retrieval as the relevant list.
    - Arabic language datasets
    - Without synonyms expansion
3. Exp3:
    - Purpose: Measuring the effect of each extractor on the relevancy of the IR system when the corpus is not semantically rich (the text's writers do not diversify their vocabularies) . to achieve this purpose we used the Blog Authorship corpus which represents young people posts, and normally those people in their posts do not diversify their vocabularies.
    - The number of inverted indexes created is 5, one from the main corpus, one from the MLSextractor extracts, one from the LSAextractor extracts, one from VSMextractor extracts, and one from the Jacextractor extracts.
    - The number of processed queries is 60, and they were selected manually.
    - Automatic relevancy assessment, the retrieval list of the main corpus inverted index as the relevant list.
    - English language datasets.

- Without synonyms expansion.

4. Exp4:
    - Purpose: the same purpose of Exp1 but with synonyms expansions.
    - The number of inverted indexes created is 5, one from the main corpus, one from the MLSextractor extracts, one from the LSAextractor extracts, one from VSMextractor extracts, and one from the Jacextractor extracts.
    - The number of processed queries is 60. The queries are mentioned in the 242 corpus.
    - Manual relevancy assessment. The relevancy assessment of the queries was mentioned in the 242 corpus and done manually by the corpus developers.
    - Arabic language datasets.
    - Synonyms expansion using the NBDV method.
5. Exp5:
    - Purpose: the same purpose of Exp2 but with synonyms expansions.
    - The number of inverted indexes created is 5, one from the main corpus, one from the MLSextractor extracts, one from the LSAextractor extracts, one from VSMextractor extracts, and one from the Jacextractor extracts.
    - The number of processed queries is 100, and they were selected manually.
    - Automatic relevancy assessment, the retrieval list of the main inverted index as the relevant list.
    - Arabic language datasets
    - Synonyms expansion using the NBDV method

The number of inverted indexes created in each experiment is five as follows:

- MCII: Main Corpus Inverted Index
- MLSECII: Inverted Index of the MLS Extracts Corpus, the inverted index created based on the extracts generated from the MLSExtractor.
- LSAECII: Inverted Index of the LSA Extracts Corpus, the inverted index created based on the extracts generated from the LSAExtractor

- VSMECII: Inverted Index of the VSM Extracts Corpus, the inverted index created based on the extracts generated from the VSMExtractor
- JACECII: Inverted Index of the Jaccard Extracts Corpus, the inverted index created based on the extracts generated from the JacExtractor

The output from each experiment includes:

1. The Similarity values between the queries and the documents found in the inverted index as follow:
    - Sim(MC, Q): the similarity between the queries and the documents appeared in MCII.
    - Sim(MLSE, Q): the similarity between the queries and the documents appeared in MLSECII.
    - Sim(LSAE, Q): the similarity between the queries and the documents appeared in LSAECII.
    - Sim(VSME, Q): the similarity between the queries and the documents appeared in VSMECII.
    - Sim(JACE, Q): the similarity between the queries and the documents appeared in JACECII.
2. The precision at each point a relevant document is retrieved (for each query and for each inverted index) as follows:
    - RP(MLS,60, MC) → MLSECII inverted index, 60 queries, main corpus retrieved set as a relevant list.

      This represents the precision at each retrieve of relevant document for 60 queries, the MLSECII is the inverted index, and the relevancy assessment is based on the retrieved set of running the IR system over the main corpus inverted index (in simple words comparing the results of running the IR system over the main corpus inverted index and over the MLSECII inverted index, and the comparison includes 60 queries).
    - RP(LSA,60, MC) → LSAECII inverted index, 60 queries, main corpus retrieved set as a relevant list.

This represents the precision at each retrieve of a relevant document for 60 queries, the LSAECII is the inverted index, and the relevancy assessment is based on the retrieved set of running the IR system over the main corpus inverted index.

- RP(VSM,60, MC) → VSMECII inverted index, 60 queries, main corpus retrieved set as a relevant list.

  This represents the precision at each retrieve of a relevant document for 60 queries, the VSMECII is the inverted index, and the relevancy assessment is based on the retrieved set of running the IR system over the main corpus inverted index.

- RP(JAC,60, MC) → JACECII inverted index, 60 queries, main corpus retrieved set as a relevant list.

  This represents the precision at each retrieve of a relevant document for 60 queries, the JACECII is the inverted index, and the relevancy assessment is based on the retrieved set of running the IR system over the main corpus inverted index.

  (In Exp 2 and exp5 the number 60 is replaced by 100 because the number of processed queries is 100, and in exp1 and Exp 4 the MC is replaced by Manual because the relevancy assessment is performed manually and already found in the dataset)

3. The final recall value of the IR system for each inverted index.
4. The final MAP value of the IR system for each inverted index.
5. The size of each inverted index.
6. The ratio of the size of the inverted index to the main corpus inverted index.
7. The Interpolated Average Precision at 11 recall points of the IR system for each inverted index.
8. The Precision-Recall curve of the IR system for each inverted index.

4.3 Experiment Results

4.3.1 MLS extraction Result

After implementing the deletion process in the automatic extraction systems described in section 3.3 and initiating the experiment outlined in section 4.2, we collected our results to make the intrinsic and extrinsic evaluation. From

the VSMExtractor, JacExtractor, LSAExtractor, and MLSExtractor and for each document, we collected the following[13]:

The automatic and manual extracts sentences(the summaries): as we said before in section 4.2.2.1 four automatic extracts will be generated, so for each document, the sentences generated from VSMExtractor, JacExtractor, LSAExtractor, MLSExtractor, and the sentences that forming the manual extracts were collected as in Table 4.8. It is an example of document 1 automatic and manual extract sentences. The document sentences were numbered sequentially during the pre-processing stage.

Table 4.8 Automatic and Manual extracts' sentences (Document 1 Essex Corpus)

Extract	Sentence id													
JACExtrator	1	2	3	5	7	11	12	13	14	15	16	17	18	19
VSAExtractor	1	2	3	5	7	9	14	15	17	18	19	20	21	22
LSAExtractor	1	2	4	5	6	9	10	12	13	15	16	18	22	24
MLSExtractor	1	2	5	7	14	15	17	18	20	22	26	29		
M1(Essex)	3	7	12	30										
M2(Essex)	2	5	7	8	13	15	17	29						
M3(Essex)	2	3	5	21					Manual Extracts					
M4(Essex)	2	5	7	8	13	15	17	29						
M5(Essex)	2	3	5	21										

The data collected in Table 4.8 is used to construct the extracts corpus that will be used as input to the indexing process instead of the main corpus. Four extracts corpus were generated, one from each extractor.

The Condensation Rate: for each extract generated from the four automatic extraction systems proposed in this research, we computed the length of the extract relative to the length of the documents (CR). See Table 4.9 as an example.

Table 4.9 Sample of Condensation Rates

doc #	JacExtractor	VSMExtractor	LSAExtractor	MLSExtractor
1	81%	56%	55%	55%
2	100%	45%	40%	36%
3	86%	92%	42%	35%

We used the CR to estimate the size of the automatic and the manual extract. This condensation rate with the RSI measure will form the containment evaluation parameters as we will see in section 5.1. The containment evaluation gives a clear indication of the effectiveness of our similarity calculations and accuracy of the extraction models.

[13] Parts of this section and its subsections are mentioned in the second paper of the "**Publications Arising from This Thesis**" section.

The RSI values: for each automatic extract generated from the four automatic extraction systems, we computed the RSI value from equation 12 in definition 10 chapter 3: Table 4.10 gives an example of the RSI values of the JacExtractor extracts with the manual extracts that are taken from Essex dataset for document 57, 135, 136.

Table 4.10 RSI Sample – JACExtractor extracts with M1, M2, M3, M4, and M5

doc#	(M1,Jac)	(M2,Jac)	(M3,Jac)	(M4,Jac)	(M5,Jac)	AVG
135	0%	31%	31%	40%	71%	35%
57	54%	67%	40%	0%	67%	45%
136	54%	33%	67%	0%	75%	45%

We collected the percentage of the automatic extracts that obtained LOWC, MODC, HIGHC, and FULLC containment with the manual extracts (the containment evaluation is presented in Definition 1, section 5.1). For example, in Table 4.11, the FULLC-Containment value between the automatic extracts generated using VSMExtractor and the manual extracts was 16%. This means that 16% of the automatic extracts produced from VSMExtractor contained all the sentences of the manual extracts.

Table 4.11 Containment Evaluation Sample

	Containment (VSMExtractor Extracts, Manual Extracts)
LOWC	19%
MODC	37%
HIGHC	29%
FULLC	16%

ROUGE 2.0 relevancy measures (AR, AP, and AF): We boosted the evaluation of our extraction models by ROUGE evaluation tool. It is used frequently to assess the quality of the automatically generated summaries. For each automatic extract generated by the four automatic extractors, the ROUGE 2.0 tool was used to find the *AR, AP, and AF* between the automatic extracts and the manual extracts that are taken from Essex and Kalimat datasets; Table 4.12 shows an example of ROUGE results for Document 1-Essex corpus. In Table 4.12, the extraction systems are the four extractors that are developed in this research plus two existing summarizers: the API summarizer and the UTF-8 SUPPORT TOOL (full description of these two summarizers appears in section 5.1.3)

Table 4.12 ROUGE Evaluation of the Automatic Extracts that were Generated for Document 1 (Essex)

Extraction System	AR	AP	AF
API summarizer	38%	82%	51%
LSA Extractor	68%	61%	64%
JacExtractor	48%	43%	45%
UTF-8 SUPPORT TOOL	31%	55%	40%
MLS Extractor	87%	62%	72%
VSM Extractor	76%	55%	64%

4.3.2 NBDV extraction Result

In the experiment of the NBDV, the following results are collected for evaluation purposes:

The verbs appeared in each run with their weights. These results are used in the evaluation to assess the ratio of verbs processed in each run to the total number of verbs in the whole corpus. The number of verbs processed is essential to determine the time complexity of the NBDV, as shown in the evaluation chapter.

Example, consider the noun "هجوم" "attack", the list of verbs and their weights computed using the OWS came as follow:

شن	بشن	شنا	يليها	صد	بشان	عنف	جرح	يلطف	رتل	خططت	تورط	ادين	زحف	نجا
Past of launch	Present of launch	Past of launch for plural	Followed by	repulsed	about	expostulate	hurt	mitigate	intone	Past of plan	mire	Past of convict	Past of crawl	Past of survive
83%	79%	54%	29%	27%	26%	19%	18%	15%	13%	11%	10%	10%	10%	9%

The maximum and minimum weights of each run: they are necessary to compute the value of the orbit range in each run (the output of applying equation 19)

Example, for the noun "هجوم", the maximum weight was 83% for the verb (launched), and the minimum weight was 9% for the verb (survived).

$$\boldsymbol{Range\ of\ values\ in\ each\ orbit} = \frac{\mathbf{83\%-9\%}}{\mathbf{16}} \mathbf{= 5\%}$$

The distribution of the verbs in the orbits: as in the following example for the noun "هجوم":

شن	بشن	شنا	يليها	صد	بشان	اعنف	جرح	يلطف	رتل	خططت	صد	تورط	ادين	زحف	نجا
83%	79%	54%	29%	27%	26%	19%	18%	15%	13%	11%	11%	10%	10%	10%	83%
Inner orbit		Second orbit	Third orbit			Fourth orbit			Outer orbit						

The set of candidate synonyms. Similar to the number of verbs, the number of nouns processed in each run is collected to estimate the complexity of the NBDV method.

Example. The candidate synonyms of the noun "هجوم":

هجمات	رئيس	ناس	اعتداءات	هجوم	حمله	عمليات	قوات	اعتداء	عمليه	عدوان	عديد	محاوله	عوده	عراق
Attacks	President	People	assaults	attack	campaign	operations	troop	Assault	operation	aggression	numerous	attempt	recurrence	Iraq

The final set of synonyms after deleting the candidate synonyms that had low similarity with the main noun: as in the following example for the noun "هجوم".

هجوم	هجمات	ناس	حمله	عمليات	اعتداء	عمليه
attack	attacks	people	campaign	operations	assault	operation

The nouns with their synonyms in one table for all runs. The outputs of 564 runs of the VSyn were collected in the following format:

Term	Syn1	Syn2	Syn3	Syn4	Syn5	Syn6	Syn7

These results are important to evaluate the accuracy (P and R) of our method. The final sets of synonyms are entered into the ROUGE evaluation tool and assessed by the manual evaluators. Table 4.13 shows a sample of our results.

Table 4.13 Results Samples of Synonyms that were Generated from our Synonyms Extraction System

Term	Syn 1	Syn 2	Syn 3	Syn 4	Syn 5	Syn 6	Syn 7
ناس people	ناس people	قوم folk	السكان residents	رسول messenger			
منطقه area	منطقه area	مدينه city	شباب youth	ولايه state	ولايات States	مناطق areas	عديد numerous
مجموعه collection	مجموعه collection	جماعه group	شكل form	وفد delegation	اكثر More	عديد numerous	عام general
ولايه state	ولايه state	منطقه area	عام general	محافظه governorate	مناطق Areas	بحريه marine	مدينه city
شارع street	شارع street	طريق road	منطقه area	وادي valley	مختلف different	ولايه states	
مليار	no synonyms generated						
شركه company	شركه company	شركات companies	وزاره ministry	منظمه organization	مؤسسه institution	عام general	مشاركه participation
عدوان aggression	عدوان aggression	رئيس president	عديد numerous	اعتداء assault	هجوم Attack	هجمات attacks	
كاتب writer	كاتب writer	مؤلف author	كتاب book	كاتبه writer (female)	عمل Work	تاريخ history	عالم world
زعيم leader	زعيم leader	زعيمه leader (female)	يوم day	غزو invasion	رئيس president	عام general	وقت time
بحث research	بحث research	دراسه study	تقديم introducing	تحقيق investigation	شباب Youth	تجديد renewal	عمل work
صور pictures	صوره picture	صور pictures	تاريخ history	تصورات perceptions			
وقت time	وقت time	شكل form	قدر destiny				
معرفه knowledge	معرفه knowledge	تحقيق investigation	وقت time	مجال domain	نور light	نفط petrol	معلومات information
مسلم Muslim	مسلم Muslim	مسلمون Muslims	مسلمين Muslims	عالم world			
طعام food	طعام food	حقول fields	حديث speech	غذاء nutriment	بروتينات proteins	ماكولات foods	

4.3.3 IR experiments Results

As described in section 4.2, five experiments were conducted to measure the effect of different models of text extraction (Jaccard, VSM, LSA, and MLS) on the size of the inverted index, and on the relevancy of the IR system. Besides that, the experiments measure the effect of the NBDV synonyms extraction method on the expansion of query terms and how this expansion affected the relevancy assessment.

To unify the way we judge the effect of the ATE models proposed in the research, we collected the following results from all the experiments:

1. The similarity values between the queries and the documents (Exp 1 to Exp 5): the similarity values between each query and the documents, which are represented in five inverted indexes, have been collected. The similarity values are used to retrieve the set of relevant documents. Table 4.14 represents the similarity values obtained in Exp 1 for the query "هندسة الحاسوب Computer Engineering" based on the five inverted indexes created in Exp 1:

Table 4.14 Similarity Values of the Query " هندسة الحاسوب " -Exp 1

MC-based Retrieval		MLS-based Retrieval		LSA-Based Retrieval		VSM-Based Retrieval		JAC-Based Retrieval	
Doc id	Sim(doc,q)	Doc id	Sim(doc,q)	Doc id	Sim(doc,q)	Doc id	Sim(doc,q)	Doc id	Sim(doc,q)
14	20%	34	21%	14	23%	14	21%	14	20%
47	19%	47	21%	178	22%	47	20%	47	19%
156	19%	14	19%	34	21%	203	19%	156	19%
203	19%	305	19%	47	21%	20	17%	203	19%
178	18%	20	18%	305	19%	164	17%	178	18%
164	17%	49	17%	20	18%	34	16%	164	17%
20	17%	164	17%	49	17%	305	16%	20	17%
49	17%	366	17%	164	17%	49	16%	49	17%
305	16%	203	9%	393	17%	366	16%	305	16%
34	16%	21	5%	366	17%	21	5%	34	16%
21	5%	178	5%	203	9%	174	5%	21	5%
174	5%	175	5%	21	6%	668	4%	174	5%
175	4%	174	5%	174	5%	393	4%	175	4%
668	4%	668	5%	668	5%	178	3%	668	4%
234	4%	234	4%	175	5%	56	3%	234	4%
696	3%	56	4%	56	3%	175	3%	696	3%

2. The sets of retrieved documents for each query that are retrieved based on the similarity values between the query and documents (Exp 1 to Exp 5): the retrieved sets of documents are sorted based on the value of similarity. The sets are then used to measure the relevancy measurements (P, R …) based on manual relevancy assessment found in the corpora described in section 4.2.1 or based on the relevant documents retrieved from MC-based retrieval. For example, the retrieved sets of documents that match the query "الاكتئاب و القلق Depression and anxiety" in the five inverted indexes that were created in Exp 2 came as in Table 4.15:

Table 4.15 Retrieved set of Documents for the query "الاكتئاب و القلق Depression and anxiety"-Exp2

MC-based retrieval	MLS-based retrieval	LSA-based retrieval	VSM-based retrieval	Jac-based retrieval
317	294	317	203	317
203	335	294	377	203
294	373	702	294	294
328	377	359	335	292
356	702	372	292	372
377	382	382	372	328
372	372	377	382	377
702	359	693	702	382
382	693	317	373	702
292			356	335
335			328	373
693			693	359
359			359	693
373				356

3. The sets of relevant documents for each query (Exp 1 to Exp 5): the relevant documents in Exp1 and Exp 4 are prepared manually in the employed datasets for example for the query "هندسة الحاسوب Computer Engineering" the set of relevant documents are 14, 24, 53, 71, 72, 75, 77, 93, 103, 178, 179, 180, 181, 182, 183, 184, 185, 186, 203, 216, 218, 219, 230. In Exp 2, Exp 3, and Exp 5, we took the retrieved documents based on the MC-based retrieval as the relevant documents. The purpose was to compare the relevancy that was achieved from the extracts based retrieval with the relevancy achieved from the MC-based retrieval. For example, the retrieved sets of documents that match the query "الاكتئاب و القلق Depression and anxiety" in the MC-based retrieval tested in Exp 2 and Exp 5 came as follow: 317, 203, 294, 328, 356, 377, 372, 702, 382, 292, 335, 693, 359, 373. And, the retrieved sets of documents that match the query "victims and criminals" in the MC-based retrieval tested in Exp 3 (the English language corpus) came as follow: 35, 16, 20, 92, 214, 74, 263, 73, 219, 75, 159, 213.

4. The relevancy assessments that include:

- The precision when each relevant document is retrieved (or AP) (Exp 1 to Exp 5): This measure is important to generate the recall-precision cure. The average precision of the MLS-based retrieval in Exp 1 are presented in Table 4.16.

Table 4.16 AP of the MLS-Based Retrieval – Exp 1

Query id	Doc Ret	R	P
3	203	6%	50%
3	178	13%	50%
3	181	19%	11%
3	182	25%	14%
3	186	31%	17%
3	179	38%	18%
3	183	44%	18%
3	180	50%	16%
3	207	56%	16%
3	184	63%	14%
3	209	69%	13%
3	215	75%	13%

- Interpolated Average Precision (Exp 1 to Exp 5): this measure traces the maximum precision at 11 recall levels, Ri = {0.0, 0.1, 0.2 , 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. After computing the Interpolated Average Precision for all the queries, the average is computed for each interval. From Exp 1, the Interpolated Average Precision values for the query "شبكات الحاسب الالي Computer Networks" with the MLS-based retrieval is mentioned in Table 4.17:

Table 4.17 Interpolated Average Precision of the Query "شبكات الحاسب الالي" with the MLS-based Retrieval – Exp 1

Recall Rank	P
r0	100%
r1	90%
r2	65%
r3	65%
r4	53%
r5	35%
r6	22%
r7	18%
r8	13%
r9	10%
r10	0%

The 90% percent appeared with r1 means that the maximum precision obtained when the recall value was greater than or equal to 10% and less than 20% is 90%. The average of the Interpolated Average Precision for all the queries (60 queries) in Exp 1 came as in Table 4.18:

Table 4.18 Interpolated Average Precision for all Queries with the MLS-based Retrieval – Exp 1 and Exp 2

Recall Rank	AP Exp1	AP Exp2
r0	51%	95%
r1	39%	98%
r2	30%	95%
r3	23%	98%
r4	18%	94%
r5	15%	90%
r6	13%	64%
r7	10%	49%
r8	6%	16%
r9	1%	0%
r10	0%	0%

Note that the Interpolated Average Precision for Exp 2 is higher than the Interpolated Average Precision for Exp 1 because in Exp 2 the relevancy is judge against the set of documents retrieved based on the main corpus inverted index, and these high values mean how close the extract-based retrieval to the main corpus retrieval. In the example above and at r5 (recall between 40% and less than 50%), the retrieved set contains 90% relevant documents of the documents retrieved when the IR system used the main corpus as the source of indexing.

- The MAP: as mentioned in section 1.3, the average precision is computed when each relevant document is retrieved. The MAP equals the average of the average precision for all the queries. The importance of this measure is related to the quality of the retrieved set, and the retrieved set depends on the IR system used. If the retrieved set contains a sufficient number of relevant documents and the relevant documents appeared at the top of the retrieved list, then the value of the MAP will be high. Note that, We unify the IR system, and we change only the source of the index, so we are not concerns about the values of the MAP; we concern about the convergence between the MAP of the MC-based retrieval and the MAP of the extract-based retrieval. This means that in case the IR system used the

main corpus inverted index or the MLS extracts inverted index, the MAP value should be convergent (high or low this is not important). For example, the MAP obtained in Exp1 and Exp 4 came as mentioned in Table 4.19:

Table 4.19 MAP Obtained in Exp1 and Exp 4

Experiment	MC-based Retrieval	MLS-based Retrieval	LSA-Based Retrieval	VSM-Based Retrieval	JAC-Based Retrieval
MAP Exp 4	40%	37%	37%	38%	40%
MAP Exp 1	40%	37%	37%	39%	39%

- The recall: The recall measures the percent of relevant retrieved documents to the total number of relevant documents. The recall is very important because we want to measure the number of the relevant documents retrieved in the MC-based retrieval and in extracts based retrieval. For example, the obtained recall values in Exp1 and Exp 4 came as mentioned in Table 4.20:

Table 4.20 Recall Obtained in Exp1 and Exp 4

Experiment	MC-based Retrieval	MLS-based Retrieval	LSA-Based Retrieval	VSM-Based Retrieval	JAC-Based Retrieval
MAP Exp 4	78%	66%	68%	75%	75%
MAP Exp 1	78%	65%	67%	74%	74%

- The ratio of the extracts inverted index size to the main corpus inverted index size: the sizes of the inverted indexes generated from the extracts that are produced from the developed extractors are measured. We measured the number of terms composing the inverted index. We did not measure the size in bytes because the accurate size is affected by the type of compression techniques and it is out of our interest. We want to see the reduction in the size of the inverted index and how this affected the relevancy. For example, the ratio of size reduction in Exp 4 came as mentioned in Table 4.21:

Table 4.21 Ratio of inverted Index Size Reduction in Exp 4

Experiment	MC-based Retrieval	MLS-based Retrieval	LSA-Based Retrieval	VSM-Based Retrieval	JAC-Based Retrieval
Ratio to the main Corpus	100%	42%	54%	68%	79%

All these results are collected from the five experiments (that are described in section 4.2) for further analysis and evaluation, as shown in chapter 5. But, it is important to note that the sizes of the inverted indexes in Exp 1, 2, 4, and 5 are identical because we used the same inverted indexes but with different constraints (such as in Exp 2 we

increased the number of queries to 100, and in Exp 4 and 5 we used the NBDV method to expand the queries). Only Exp 3 has different sizes of the inverted indexes because they represent a new dataset (English Language dataset).

The experiment and results chapter comprised seven experiments (two text extraction experiments and five IR experiments). From these experiments, the generated results in forms of text summaries, synonyms, and retrieved lists of documents have been collected for further assessment. In the evaluation chapter, the evaluation process of the collected results will be initiated. This evaluation involves the proposed containment evaluation and the standard text mining evaluation tool (ROUGE) and metrics (R, P, f-score, MAP, AP, AR).

CHAPTER 5 EVALUATION and DISCUSSION

After collecting the results as described in chapter 4, we evaluated them using the intrinsic and extrinsic approaches. The intrinsic approach is used to evaluate the MLS and NBDV extractions. The intrinsic approach evaluates the accuracy of the answer set produced by the summarization systems. Mainly we used the RSI with the CR, and the ROUGE tool to evaluate the results obtained in the four text extractors that are developed in section 3.3. And, we used the ROUGE tool and the manual evaluation to evaluate the NBDV synonyms extraction methods. The Extrinsic evaluation for the ATS methods developed in our research is performed by employing the text extractors and the NBDV synonyms extractor in an information retrieval system.

5.1 ATE evaluation and analysis

In section 3.3.2, we explained that we built the LSAExtractor, VSMExtractor, and JacExtractor to compare them with the MLSExtractor. The evaluation of the quality of the automatically generated extracts was performed using the values of AR and AP generated from the ROUGE 2.0 Evaluation tool, and the values of RSI integrated with the values of CR. After collecting the results from the four automatic extractors, we compare the results of our automatic extractors with the results generated from two multilingual automatic extraction systems, UTF-8 SUPPORT TOOL, and Text Summarization API. The same documents manipulated by our extractor were processed using the UTF-8 SUPPORT TOOL and API, and the recall, precision, and f-score values generated by the ROUGE tool were collected. The final step of our evaluation was to analyze the time consumed by each extraction system and to measure the enhancement achieved by using the MLS method on both the matrix reduction and the number of runs of the LSA procedure[14].

5.1.1 The Containment Evaluation

The RSI, which is defined in section 3.3.2 stage 3, was used during the evaluation. The RSI measures the percent of complete sentences that are shared between two extracts, but to make the RSI more significant, we categorized the RSI values in ranges, as shown in definition 1.

[14] Parts of this section and its subsections are mentioned in the second paper of the "**Publications Arising from This Thesis**" section

Definition 1: Let RSI_x be the value of RSI between the automatic and manual extracts:

LOWC containment occur if $RSI_x < 50\%$.

MODC containment occurs if $50\% <= RSI_x < 75\%$.

HIGHC containment occurs if $75\% <= RSI_x < 100\%$

FULLC containment occurs if $RSI_x = 100\%$.

Example: For document 6 in Essex corpus, the RSI (M1 extract, MLS extract) was 100%; this means that all the sentences found in the manual extract M1 appeared in the MLS extract and this yields FULLC containment.

5.1.1.1 RSI Findings

We used RSI to measure the percentage of containment of manual extracts in the automatic extracts generated from our automatic extractors. Figures 5.1 – 5.4 show the percentage of containment of the manual extracts taken from Essex corpus in the automatic extracts generated automatically by the four extractors developed in this research (five manual extracts for each document).

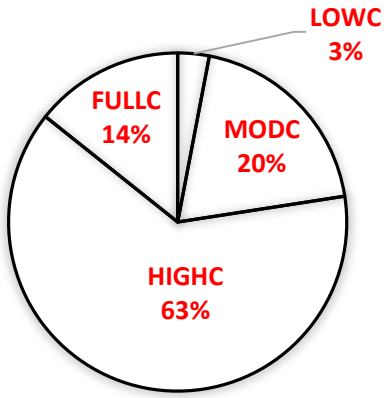


Figure 5.1 The Containment evaluation of JacExtractor Extracts

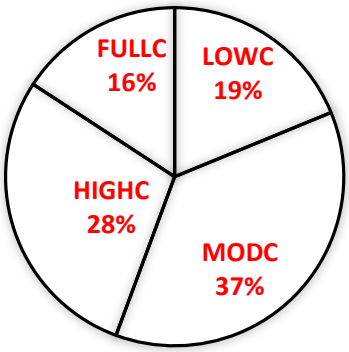


Figure 5.2 The Containment evaluation of VSMExtractor Extracts

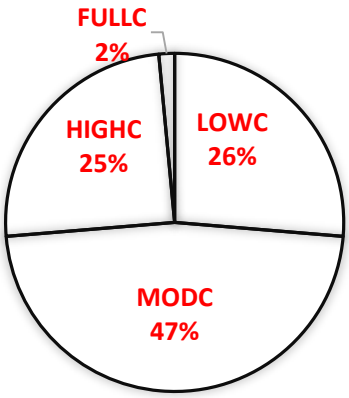


Figure 5.3 The Containment evaluation of LSAExtractor Extracts

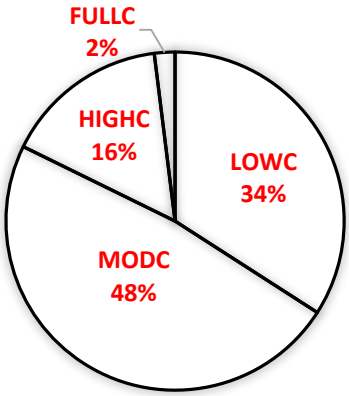


Figure 5.4 The Containment evaluation of MLSExtractor Extract

The final results of the containment evaluations are summarized in Figures 5.1- 5.4, we chose the pie chart graph in these figures because we see each containment level as a sector and the pie chart can easily show the ratio of each sector. From Figures 5.1 - 5.4, we note the following:

The LOWC containment means that less than half of the sentences found manually appeared in the automatic extract. The obtained LOWC containment values by our extraction systems were low, ranging from 3% in the JacExtractor and 34% in the MLSExtractor. The LOWC containment value gives a good indication of the extracting accuracy because their values strongly determine the contents of the automatic extract. In fact, when the value of the LOWC containment is 34% in the MLSExtractor extracts, this means the remaining 66% of the manual extract sentences appeared in the automatic extracts generated by MLS extractor. In other words, the vast majority of the automatic extracts contain more than 50% of the sentences of the manual extracts.

The FULLC and HIGHC containment values for both JacExtractor and VSMExtractor were high and greater than their corresponding values in the LSAExtractor and the MLSExtractor. These two levels of containment mean that the automatic and manual extracts shared more than 75% of the sentence, and even if their values are not significant for the MLSExtractor this will not shock us because, at 42% CR, we don't expect the FULLC and HIGHC to be high.

The MODC plus HIGHC Containment (the containment of greater than 50% and less than 100% of the manual extracts in automatic extracts) dominated the largest sector for the four extraction systems. The value of the MODC-plus HIGHC Containment was 85% for VSMExtractor, 83% for JacExtractor, 72% for LSAExtractor, and 64% for the MLSExtractor. Note that a significant ratio of the automatic extracts that were generated from the four extractors succeeded in sharing more 50% of the manual extract sentences.

The value of FULLC plus HIGHC containments (the containment of greater than 75%) for the JacExtractor and VSMExtractor was high (66% and 44%). Whereas, the FULLC and HIGHC containment value was acceptable (27%) for the LSAExtractor and low for the MLSExtractor (18%). The percent of 100% containment (FULLC-Containment) was noticeable for JacExtractor and VSMExtractor 14% and 16% respectively, but for the MLSExtractor and the LSAExtractor, the FULLC containment appeared in two cases only and showed 2% FULLC containment. However, the FULLC containment is the most significant level in the containment evaluation and its value resembles the recall value in the ROUGE tool, so to give the FULLC more considerable value, we should link it with the final CR for each extractor.

5.1.1.2 Integrating RSI and CR

The extraction systems built in this investigation produces variable size extracts, which means that the size of the extract is not constrained by a specific ratio or number of terms. Thus, it is necessary to compute the average CR rate for the extracts that are generated from each extractor.

If we consider the FULLC and HIGHC containment as the optimal results, we can order the accuracy of the automatic extractors as follows: The VSMExtractor comes at the top followed by the JacExtractor followed by the LSAExtractor, and the MLSExtractor came at the end. But, The RSI containment as a measure of evaluation is helpful if we combine it with the condensation rate, which is an important measure used to test the quality of the machine extracts.

If we integrated the results showed in Figures 5.1 – 5.4 with Figure 5.5, we find that the RSI containment values for both the JacExtractor and the VSMExtractor extracts were obtained at a high value of CR (79% and 68% respectively), and this explains why the JacExtractor and the VSMExtractor extracts contained more sentences of the manual extracts than LSAExtractor and MLSExtractor extracts. The automatic extracts that are generated from the JacExtractor and the VSMExtractor have large sizes and the amount of reduction is inconsiderable. Comparing with the average CR of the LSAExtractor and MLSExtractor, we can find that the LSAExtractor system surpassed the JacExtractor and the VSMExtractor because it obtained high containment value in reasonable CR value (54%).

Regarding the MLSExtractor extract, it achieved reasonable containment assessment (66%), and it succeeded in removing 58% of the original text. Figure 5.5 reveals the major drawback of VSMExtractor and JacExtractor systems. The CR values are impractical, which means that these automatic text extraction systems did not cancel a generous portion of the original text. On the other hand, LSAExtractor and the MLSExtractor systems roughly decreased the text size to the half and obtained significant RSI values.

After combining the CR with the containment findings, we can rearrange the performance of the four extraction systems and put the MLSExtractor and LSAExtractor at the top. The choice between the MLSExtarctor and the LSAExtractor should consider the efficiency of the two systems; the efficiency analysis of the systems is presented in section 5.1.4.

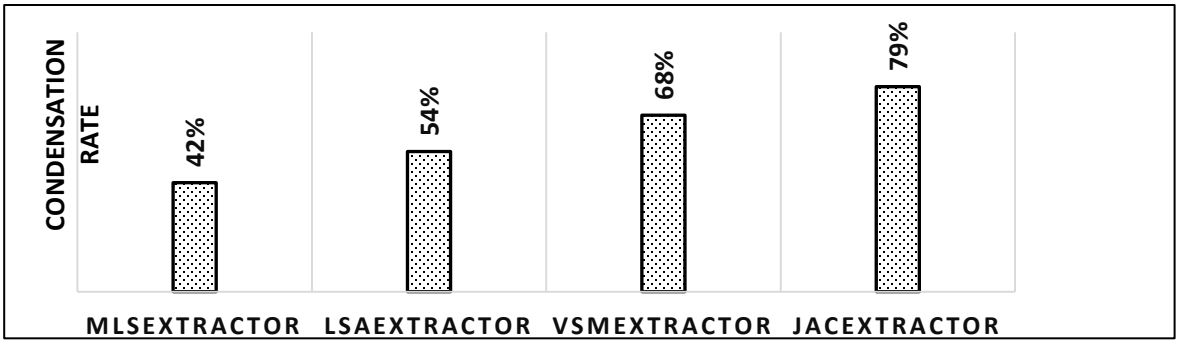


Figure 5.5 the Average Condensation Rates for the Extracts Generated by the Four Automatic Extractors

5.1.2 Evaluation Using Rouge Evaluation Tool

We used the ROUGE 2.0 evaluation tool to measure the similarity between the automatically generated extracts and the manual reference extracts found in Essex and Kalimat datasets. Also, to make the ROUGE evaluation more helpful in assessing the quality of the produced extracts, we connecting our ROUGE results with the average CR. The ROUGE evaluation was performed to boost the results that were obtained in the containment evaluation.

5.1.2.1 ROUGE Evaluation over Essex Corpus.

We used the same documents tested in the Containment evaluation, and we separated the documents into six datasets depending on their subject; the subjects include education, art and music, environment, finance, health care, and politics. We collected the recall, precision, and f-score for each dataset and all the datasets together. The datasets have a different number of documents, the smallest one contains five documents, and the largest dataset contains 30 documents. The purpose of the separation of the documents to smaller data sets is to evaluate the automatic extracts with a variance number of documents.

Table 5.1 and Figure 5.6 represent the final results that were obtained for the ROUGE AR, AP, and AF. Figure 5.6 shows the average ROUGE results and Table 5.1 shows the detailed ROUGE results for the six datasets. From Table 5.1, the MLSExtractor extracts obtained the lowest recall value. The average recall value of the MLSExtractor ranged from 38% in dataset 2 with 10 documents to 57% in dataset 4 with 14 documents, and for the whole corpus, the average recall for the MLSExtractor showed in Figure 5.6 was 48%. The average precision was significant and reaches 50% for Health care documents, with AP for all the datasets equals 40%.

Table 5.1 ROUGE Results for Six Datasets

	AR	AP	AF
Education (5 docs)			
JacExtractor	63%	25%	36%
VSMExtractor	61%	26%	36%
LSAExtractor	50%	21%	29%
MLSExtractor	54%	29%	37%
Art – Music (10 docs)			
JacExtractor	59%	27%	37%
VSMExtractor	50%	29%	37%
LSAExtractor	69%	38%	45%
MLSExtractor	38%	31%	33%
Environment (30 docs)			
JacExtractor	70%	36%	46%
VSMExtractor	68%	37%	47%
LSAExtractor	68%	42%	51%
MLSExtractor	55%	41%	46%
Finance (14 docs)			
JacExtractor	59%	33%	43%
VSMExtractor	72%	32%	44%
LSAExtractor	63%	41%	48%
MLSExtractor	57%	43%	44%
Health (12 docs)			
JacExtractor	78%	42%	54%
VSMExtractor	68%	39%	49%
LSAExtractor	58%	53%	54%
MLSExtractor	45%	50%	46%
Politics (14 docs)			
JacExtractor	75%	29%	42%
VSMExtractor	65%	29%	39%
LSAExtractor	51%	46%	46%
MLSExtractor	46%	41%	40%

From Figures 5.5 and 5.6, the MLSExtractor extracts shared 48% with the manual extracts at 42% CR value, whereas the JacExtractor achieved 70% recall but at 79% condensation rate. Note that the higher condensation rate means the system failed to omit a large portion of the text, which means that we should discard the recall values for both the VSMExtractor and the JacExtractor. The average recall of LSAExtractor was less than the average recall of VSMExtractor and JacExtractor because both JacExtractor and VSMExtractor failed to remove a reasonable part of the text and this appeared clearly from their CR values. The CR value of the LSAExtractor is

less than the CR value of the VSMExtractor by 14% and less than the CR value of the JacExtractor by 25% (see Figure 5.5).

Regarding the precision values, the MLSExtractor obtained average precision higher than the average precision of the VSMExtractor extracts and the JacExtractor extracts, and the average precision of the MLSExtractor was very close to the average precision of the LSAExtractor extracts (see Figure 5.6). Note that the average precision value of the VSMExtractor and JacExtractor are 33%, 33% with average CR values equal to 68% and 79%, respectively. The large size of the automatic extracts that were generated from the VSMExtractor and the JacExtractor participated in obtaining low precision value.

Comparing the ROUGE results of the LSAExtractor and the MLSExtractor, we found that the obtained ROUGE results from the MLSExtractor were very close to the LSAExtractor results, the average precision for the MLSExtractor is less than the average precision for the LSAExtractor by 1%, and the average recall for the MLSExtractor are less than average recall for the LSAExtractor by 12%. However, the MLS extraction achieved those ROUGE results at 42% CR rate (less than the LSAExtractor CR by 12%), which gives the MLS extraction advantage over the other automatic extraction.

5.1.2.2 ROUGE Evaluation over Kalimat Corpus.

In this subsection, we repeated the ROUGE evaluation, but with different datasets, the purpose was to boost the results obtained over the Essex dataset. We established the same experiment over the Kalimat data corpus. The same relevancy measures (AP, AR, and AF) were collected.

Figure 5.7 shows the final and average ROUGE results over Kalimat, the achieved average recall, average precision, and average f-score values were higher than the ones appearing in Figure 5.6 because, in Kalimat corpus, we have only one manual reference summary while in Essex we have five manual reference summaries and this reduces the number of comparisons performed by ROUGE.

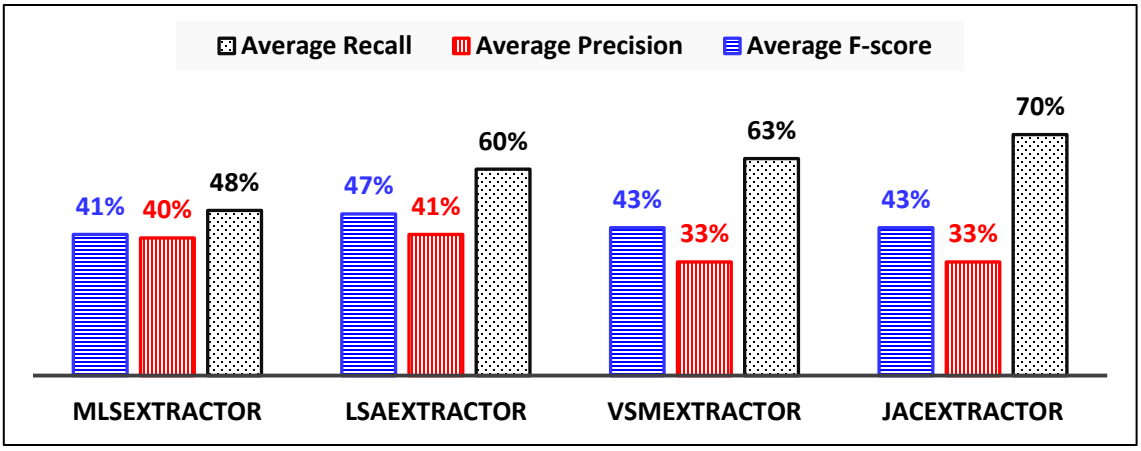


Figure 5.6 ROUGE Results for the Four Automatic Extraction Systems (Essex and 242-Document corpus)

Figure 5.6 that represents the ROUGE results over Essex corpus and Figure 5.7 that represents the ROUGE results over Kalimat corpus present convergent trends. Indeed, we neglected the precision and recall values related to the JacExtractor and the VSMExtractor because the corresponding CR was insignificant. In Figure 5.7, the precision values between the LSAExtractor and the MSLExtractor are convergent, and the MLSExtractor obtained the highest value (70%). The recall of the LSAExtractor was higher than the recall of the MLSExtractor by 15% but with 12% difference in the CR (see Figure 5.5). However, 57% of recall is considered reasonable if we connect that with the CR and with the amount of reduction obtained in both, the original matrix and the number of runs of the LSA similarity function significant (as we will see in Figure 5.14 and Figure 5.12).

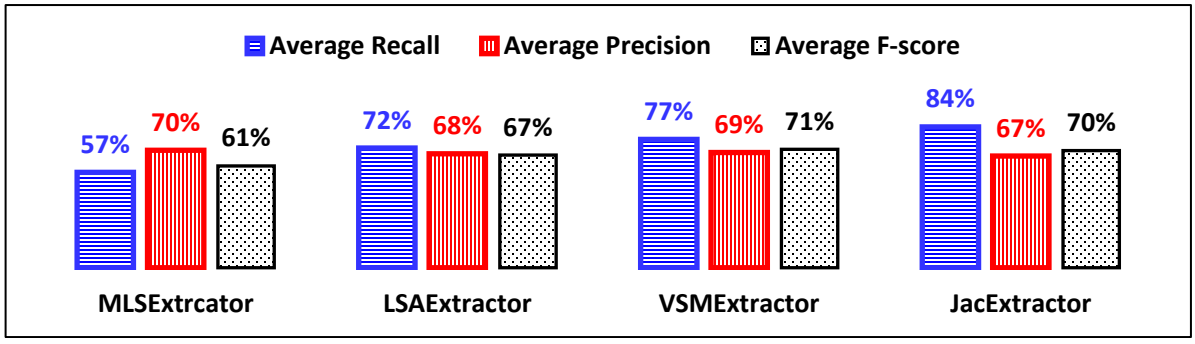


Figure 5.7 ROUGE Results for the Four Automatic Extraction Systems (Kalimat corpus).

5.1.3 Comparing MLS extract with existing Automatic Extraction Systems

In sections 5.1.1 and 5.1.2, the evaluation of the MLS extraction extract was performed by comparing the automatic extracts with gold extracts taken from Essex and Kalimat datasets. However, it important to compare the precision and recall of the MLS automatic extracts with the automatic extracts that are generated from well-known automatic text extractors. We chose two multilingual automatic extraction systems, UTF-8 SUPPORT TOOL[15], and Text Summarization API[16].

Text Summarization API is an online software service to extract the salients sentences from a text document. The user must determine the extract's number of sentences. The API tool for text extraction is based on a machine learning approach and can be used on different platforms.

UTF-8 SUPPORT TOOL is an online software service to extract the salients sentences from a text document. The user must determine the extract's ratio of sentences. This text extraction tool is a feature-based summarization system that examines certain sentence features during the summarization process, such as the sentence position, the centroid, keywords, and common subsequences. It is a single and multi-document summarization tool that supports multilingual summarization.

The reason for choosing the UTF-8 SUPPORT TOOL and the API text extractor is the language dependency; they are statistical approaches of text summarization and can be applied to the Arabic text. The same documents used during the evaluation of our automatic extraction systems were summarized using the API and the UTF-8 SUPPORT tools. The ROUGE evolution tool is used to evaluate the generated extracts against the manual extracts, and the recall and precision values were collected. We used 40% CR for both UTF-8 SUPPORT TOOL and API Summarizers because we obtained this value of CR for the MLS extraction and this experiment compares the extracts generated from those two automatic systems with our MLS extracts.

The average recall and average precision of the extracts that were generated from the MLSExtractor, UTF-8 SUPPORT TOOL, and the API text extractor are presented in Figure 5.8. Regarding the average recall, the MLS automatic extractor obtained the highest value(48%). Regarding the average precision, the results were convergent,

[15] https://www.tools4noobs.com/summarize/
[16] http://textsummarization.net/

the MLS extraction recorded 40% precision, which represents 1% improvement over the UTF-8 SUPPORT TOOL precision, and 4% over the API extracts' precision. Also, the average f-score of the MLS extraction was higher than the average f-scores of the UTF-8 SUPPORT TOOL and the API text extractor.

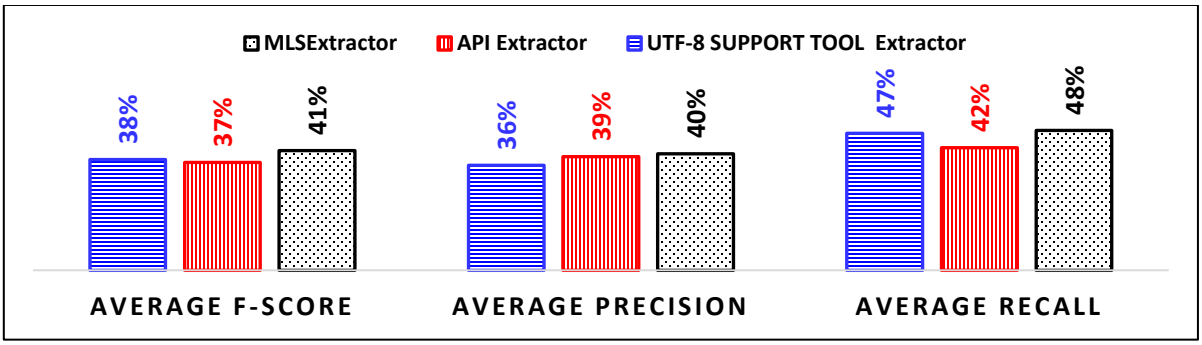


Figure 5.8 Average Recall and Precision Values for MLS, API, and UTF-8 SUPPORT TOOL Extractors.

During the evaluation of the UTF-8 SUPPORT TOOL and the API text extractor, we see that the variances in recall and precision values from document to document are noticeable. Figure 5.9.a presents the fluctuation on the recall values for the first 86 documents. Figure 5.9.b presents the fluctuation on the precision values for the first 86 documents. For UTF-8 SUPPORT TOOL and API extractors, the lines that represent the recall and precision went up and down and far from their arithmetic mean that appear in Figure 5.6. Whereas, the lines that represent the recall and precision of the MLS extraction showed more stability, and the individual values of recall and precision remain close to their means. We calculated the variances by computing the standard deviation of the resulted recall and precision values for the three systems. We employed the standard deviation to measure how close the individual values of recall and precision to their mean. If the value of the standard deviation is high, this means that the values are disparate. The standard deviation values came as followed:

The standard deviation of the MLS precision = 9%
The standard deviation of the API precision) = 16%
The standard deviation of the UTF precision = 10%

The standard deviation of the MLS recall = 14%
The standard deviation of the API recall) = 18%
The standard deviation of the UTF recall = 20%

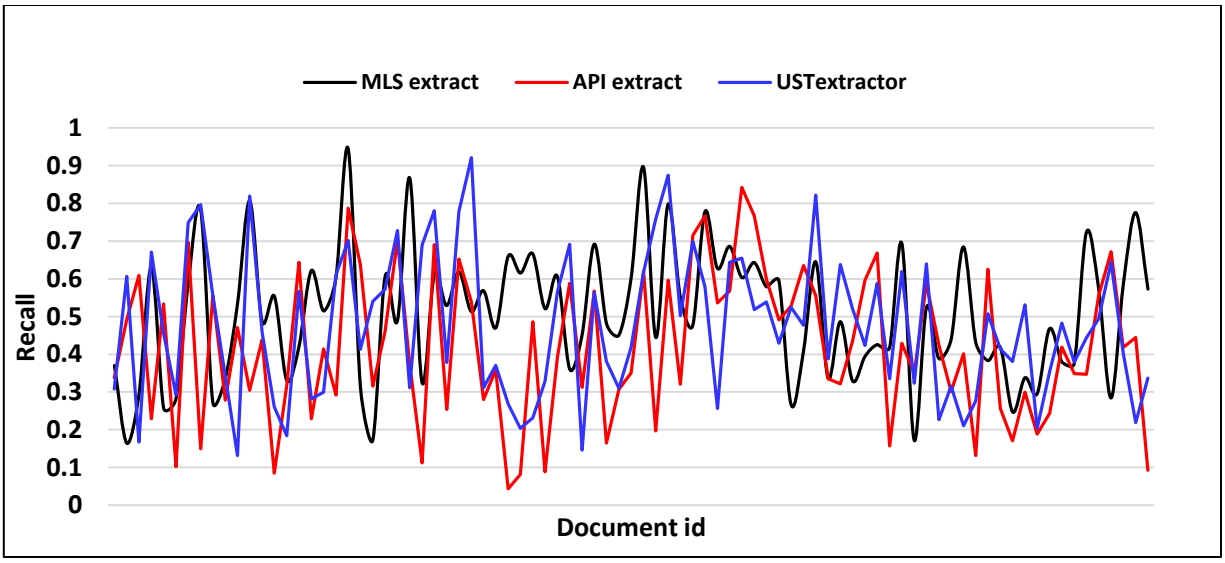


Figure 5.9.a Recall Values for MLS, API, and UTF-8 SUPPORT TOOL Extracts.

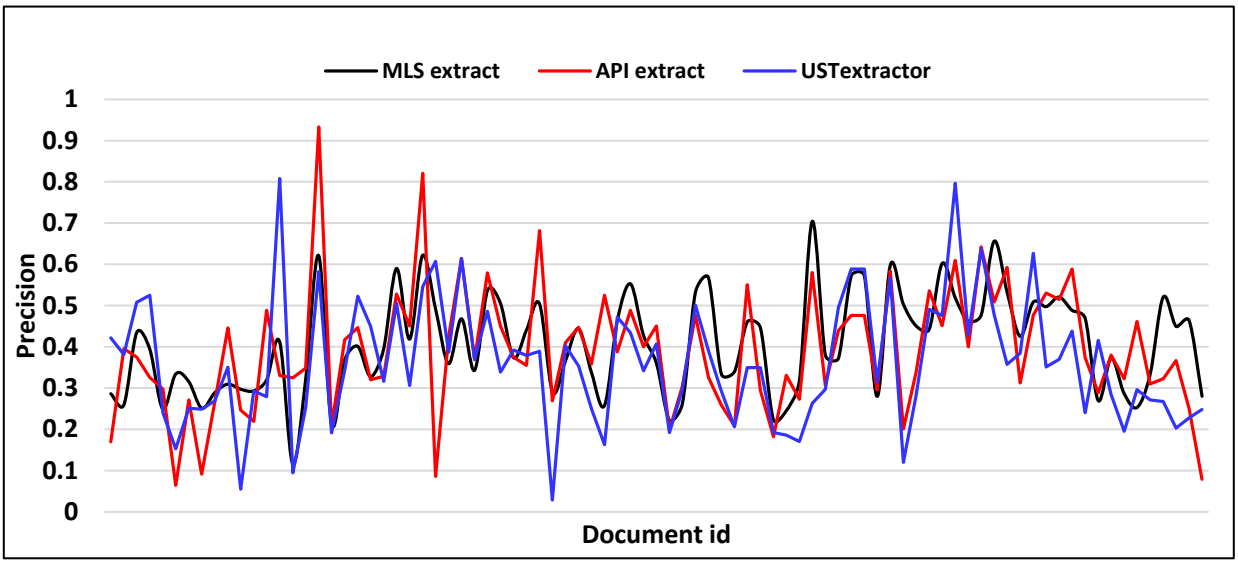


Figure 5.9.b Precision Values for MLS, API, and UTF-8 SUPPORT TOOL Extracts.

5.1.4 MLS Time Complexity Analysis

Determining the time complexity of the Jaccard coefficient and the vector space model is important because it gives a direct indication of the time complexity of the MLS model. The Jaccard Coefficient, which represents the first processing layer in the MLS, estimates the similarity between two sentences by considering the number of shared terms between them, if we have n sentences in a given document and m terms in each sentence, the overall time

complexity is $O(mn)$**.** The vector space model starts by calculating the terms' scores of weights; in a corpus of t terms and d documents, the VSM, which represents the second processing layer in the MLS, performs the weighting phase in $O(td)$ time complexity. Also, the VSM model involves the cosine similarity computations between the vectors that represent the sentences in a document. With n sentences in a document; the VSM needs $O(n\hat{}2)$ ***to compute the cosine similarity***. The total complexity of the VSM is $O(td) + O(n\hat{}2)$ and n is less than or equal t and d is normally not a small number, this yields $O(td)$ complexity. The complexity analysis of the LSA procedure, which represents the final processing layer in the MLS, is mentioned in (Wang, Xu, & Craswell, 2013) and showed expensive time penalty ($O(\min\{t^2d\},\{td^2\})$).

The relevancy evaluation of the classical LSA extraction was promising, as shown in the containment and ROUGE evaluation subsections. However. If we have n sentences in a document, this means that we need to run the LSA procedure in classical LSA extraction $n\left(\frac{n-1}{2}\right)$ times (n -1 for the first sentence plus n-2 for the second sentence, and so on). For example, for a document that contains 10 sentences, we need to run the LSA 45 times, and for a document that contains 100 sentences, we need to call the LSA procedure 4950 times. This produces a huge number of calling times of the LSA procedure in the classical LSA text extraction, especially for large documents. The main aim of MLS text extraction is to improve the efficiency in terms of time and space, so the LSA procedure should be called in minimum and only for complicated cases.

To see if the MLS reduced the number of calling times of the LSA procedure, we traced the number of times the MLS called the LSA procedure by considering the number of calls of the Jaccard and VSM procedures. Because, according to MLS processing hierarchy, if the Jaccard procedure in the first layer and the VSM procedure in the second were performed, the LSA procedure would not be executed. For example, The LSAExtractor executed the LSA procedure for document number one 435 times, whereas the MLSExtractor executed the LSA process for the same document 194 times, because 241 runs of the Jaccard and VSM procedures were recorded.

The tracing results of the number of LSA calling times were lucrative, we found that the number of calling times of the LSA procedure decreased to 52% from the original calling times in the classical LSA extraction. Figure 5.10 presents a comparison in the number times we called the LSA procedure in the MLS and LSA extractions for the

first 133 documents processed in our experiments. Figure 5.10 revealed that the LSAExtractor executed the LSA procedure 23500 times and the MLSExtractor executed the LSA procedure 11438 times (52% reduction)

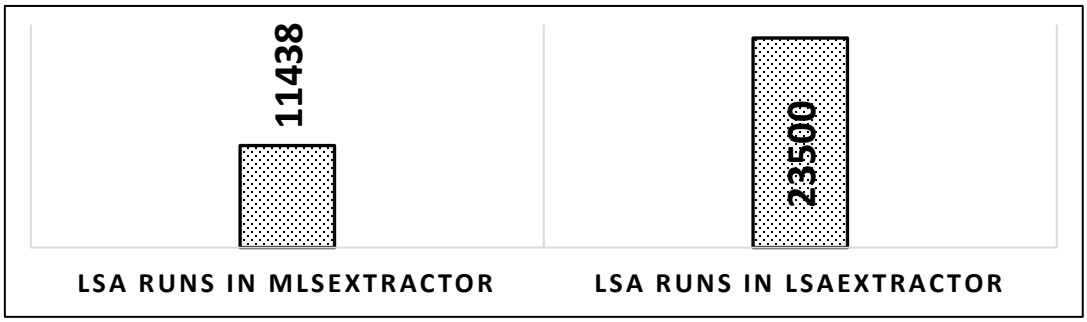


Figure 5.10 LSA Number of Runs Using LSAExtractor and MLSExtractor for 133 Documents

The original matrix dimensions are other important aspects; they affected the space and time complexity of the extraction process. The time and space complexity increase as the dimensions of the original matrix increase. As described in the methodology chapter, the MLS extraction processes the matrix by using the Jaccard and the VSM models and then transfers the remaining sentences to the SVD to perform the required matrix factorization process. The Jaccard and VSM processing of the original matrix will reduce the dimensions and produces a smaller inputted matrix to the SVD. Figure 5.11 compares the dimensions of the original matrix in the MLS based extraction and the classical LSA based extraction. Figure 5.11.a presents the reduction in the number of terms, which represents the number of rows in the original matrix, and Figure 5.11.b presents the reduction in the number of sentences, which represents the number of columns in the original matrix. To obtain these two figures (Figure 5.11.a, 5.11.b), we collected the values of $i \times j$ from the original matrix (the inputs to the SVD in classical LSA extraction) and i_{red} and j_{red} from the reduced matrix (the input to the SVD in MLS extraction). In Figure 5.11.a, the horizontal axis represents the document's id and the vertical axis represents the number of terms (j in the original matrix, j_{red} in the reduced matrix). In Figure 5.11.b, the horizontal axis represents the document's id and the vertical axis represents the number of sentences (i in the original matrix, i_{red} in the reduced matrix). The documents in both figures were sorted according to their size from left to right (the largest on the left). Note that the large documents, which are located on the left-hand side of both figures, obtained an important reduction in both the values of i and j. Also, note that the small documents, which produce small values of i and j and are located on the right-hand side, obtained a low reduction, which seems logical because for small documents, the dimensions reduction is not necessary.

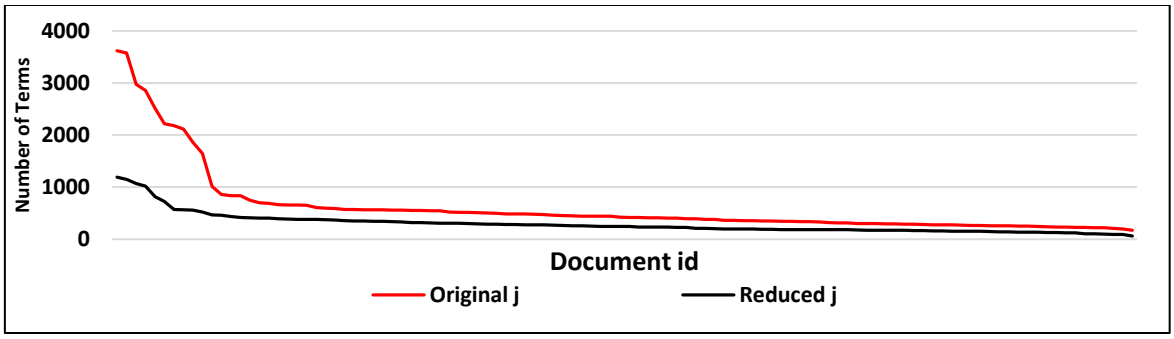


Figure 5.11.a the Trend of the Original Number of Terms that were Processed by the LSAExtractor and the Reduced Number of Terms that were Processed by MLSExtractor.

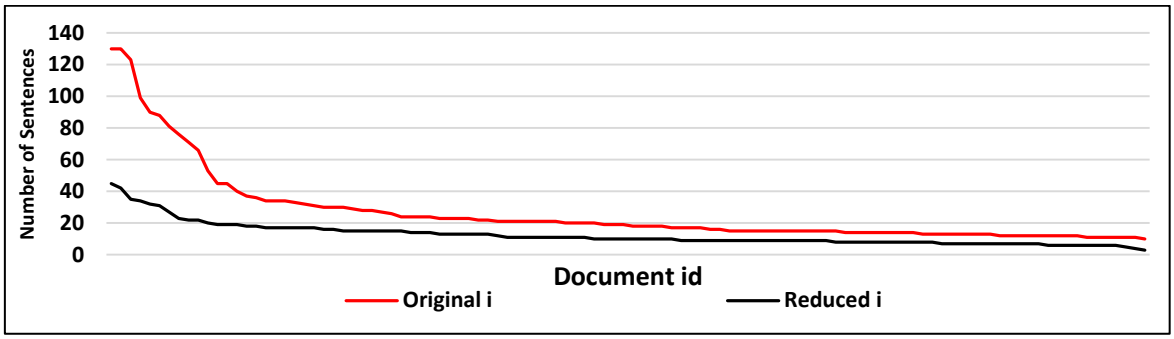


Figure 5.11.b the Trend of the Original Number of Sentences that were Processed by the LSAExtractor and the Reduced Number of Sentences that were Processed by MLSExtractor

Table 5.2 presents the final amount of reduction for all the documents that were processed in our experiment (Kalimat and Essex documents). The final average of reduction on the dimensions of the original matrix is significant for large documents (65% reduction in j dimension, and 66% reduction in i dimension).

Table 5.2 the Ratio of the Reduction Obtained by MLS

	(Average(j_{red}))	(Average(i_{red}))
Documents with number of sentences > 10	43%	45%
Documents with number of sentences > 40	65%	66%

5.2 NBDV evaluation and analysis

The evaluation includes the assessment of the precision and recall values obtained in the experiment and the time complexity necessary to run the NBDV method. But, before evaluating them, the size of the answer set produced by the VSyn system should be investigated.

5.2.1 The Size of the Answer Set Evaluation

The size of the answer set is important because if the number of generated synonyms is always high this means the parameters specified in the description of the NBDV method are not robust and cannot control the synonyms retrieval process, and if the number of synonyms retrieval is always low this means that the parameters cannot establish a real semantic relationship between the noun and the candidate synonyms.

The number of synonyms generated for each noun was statistically determined and divided into two categories, less than or equal 3 and greater than 3. On average 20% of the nouns gained between one to three synonyms, and 70% of the nouns gained more than 3 synonyms (the remaining 10% of nouns gained 0 synonyms). Figure 5.12 shows the percent of the corpus nouns that gained greater than a certain number of synonyms. For example, from Figure 5.12, 20% (110 nouns out of 564 nouns experimented) of the corpus nouns gained at least one synonym and at most three synonyms, and 70% of the corpus nouns gained at least four synonyms and at most 7. Figure 5.13 revealed that from 564 nouns tested, 90% gained at least one synonym, and 70% gained at least three synonyms. Almost, the output answer is not empty, and the VSyn system that was built based on the NBDV method succeeded in returning the reasonable number of synonyms.

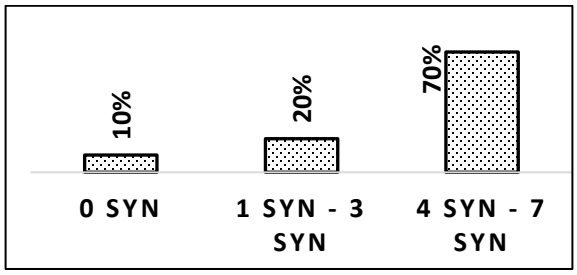


Figure 5.12 The Ratio of Nouns that gained 0, 1-3, and 4-7 Synonyms.

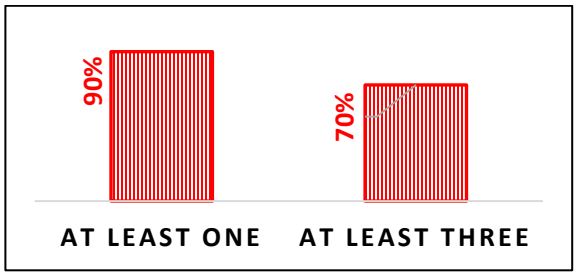


Figure 5.13 Accumulative Ratio of Nouns that Gained more than 1 and more than 3 Synonyms

5.2.2 NBDV Accuracy Assessment Evaluation

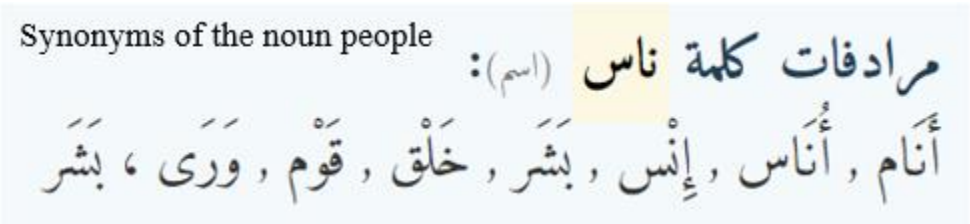
Both recall and precision are used intensely to assess the accuracy of the Natural Language Processing and Information Retrieval applications. The precision indicates the ratio of correctness relative to the answer set, while the recall gives a broad view and estimates the accuracy relative to the typical answer. However, in such kind of text mining applications, the determination of the typical answer is practically hard, but we can assume that the synonyms set found in the base dictionaries are the ideal answer and prepare our calculation accordingly. Also, Arabic language experts and speakers were hired to take their opinions in the accuracy of the generated answer

set. In the manual evaluation, it is hard to ask the experts and speakers to find the recall because this needs large effort from them to find the optimal synonyms set, so from the manual evaluation, the precision is the only relevancy measure collected. Brief descriptions of the figures and tables appear in this section are as following:

- Figure 5.14 shows the recall and precision (Almaany dictionary as an optimal answer).
- Figure 5.15 shows the recall and precision (Google Translate synonyms set as an optimal answer).
- Figure 5.16 Compares the Average recalls in the case of using Almaany as a base of evaluation and in the case of using Google Translate as a base of evaluation.
- Figure 5.17 Compares the Average precisions in the case of using Almaany as a base of evaluation and in the case of using Google Translate as a base of evaluation.
- Table 5.3 shows the average precision for the experts and speakers manual evaluation
- Table 5.4 shows the final average precision and average recall using the manual and automatic evaluation strategies

5.2.2.1 Almaany-Based Evaluation

Almaany online Dictionary contains the meanings, synonyms, and antonyms of the Arabic language words. Almaany is a pioneering online tool that composes a database taken from a set of famous Arabic dictionaries including “Lesan Alarab لسان العرب”, ”Alraa’d الرائد”, ”Alwaseet الوسيط”, ”Alghany الغني”, ”Modern Arabic Language اللغة العربية المعاصر”, and “Aljaam’a الجامع”. The output of Almaany dictionary looks like the following picture:



The answer set of Almaany usually contains repeated words, so a simple preprocessing stage to remove the duplicates was performed before the evaluation process. For example, the answer set for the noun “فوز winning” includes the following synonyms:

انتصار، انتصار، انتصار، ظفر، ظفر، ظهور، غلبه، غلبه، غلبه، غلب، غلب، فتح، فلاح، مفازه، منجاه، نجاح، نصر

Note that the Arabic noun "انتصار triumph" repeated three times, the noun "غلبة predominance" repeated five times.

The group of nouns was inserted into Almaany dictionary, and the generated synonyms were collected in the following format.

Synonyms Generated from Almaany	Synonym Generated from our automatic synonyms finder	R	P

For example, for the noun "الناس people", the results were:

Synonyms from Almaany	Synonym from our automatic synonyms finder	R	P
انام ، اناس ، انس ، بشر ، خلق ، قوم ، وري ، بشر،ساكن	قوم ناس السكان رسول	3/9 = 33%	3/4= 75%

The precision and recall results of Almaany based evaluation were summarized in Figure 5.14. Figure 5.14 reveals that the precision was higher than the recall, this means that among the returned set of synonyms for a specific noun, the accuracy was significant (the number of correct synonyms to the number of automatically generated synonyms was high), but the system did not return the sufficient number of synonyms found in the Arabic language (the number of correct synonyms to the number of synonyms found in Almaany was low). Also, the majority of the precision's values are confined between 40% and 50% (the average is 46% see Table 5.4), whereas the recall values fluctuated from 5% to 100%, and we explained that by the fact that Almaany reference sets contain one or two synonyms for certain nouns and contain more than 30 synonyms for other nouns.

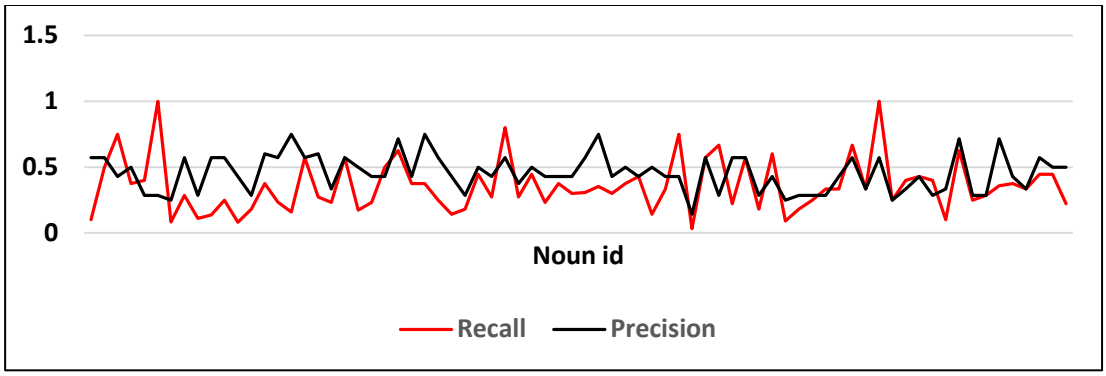


Figure 5.14 Recall and Precision – Almaany-based Dictionary

5.2.2.2 Google Translate-based Evaluation

The problem we faced in Almaany-Based Evaluation was the contents of the synonyms sets taken from the Almaany Dictionary. As described in the previous chapter, Almaany combines six well-known and ancient dictionaries, and most of its vocabularies are not in use in today's newspapers and journals (the source of our Dataset). For example, in the synonyms list of the term "people" mentioned above ("ناس" people), the synonym "وَرَى" is not in use in today's language, and the synonym "خَلْق" is not used in these days to refer to "people" (it refers to anything created by God). Therefore, the search for a new source of the synonyms that reflect modern Arabic was necessary. Google Translate was the result of this search because it gives a list of modern synonyms for any Arabic term being translated. For example, the term used in our example above ("ناس" people) has the following synonyms in Google Translate: (folk قوم, people ناس, populationسكان , society مجتمع, family أسرة). Google Translate uses statistical machine translation approach to translate the Arabic language, and it collects the meaning and synonyms from a massive number of Arabic articles found on the internet at the time of translation. (Most of these articles are written in modern Arabic vocabularies).

Figure 5.15 shows the precision and recall curve. In Figure 5.15, the values of the precision and the recall are convergent, and the recall is more stable because the sets of synonyms appeared in Google Translate (the base of the comparison) are smaller than their analog in Almaany dictionary and roughly contain the modern Arabic terms. This also affected the final average precision and average recall. Table 5.4 shows that the obtained average precision and average recall are higher than their corresponding values based on Almaany dictionary and the reason for that is the language type used in Google Translates, Google Translate uses the same language used in

our corpus, so we have the thought that the average precision and recall obtained in Google based evaluation are more significant than the results obtained in the previous subsection.

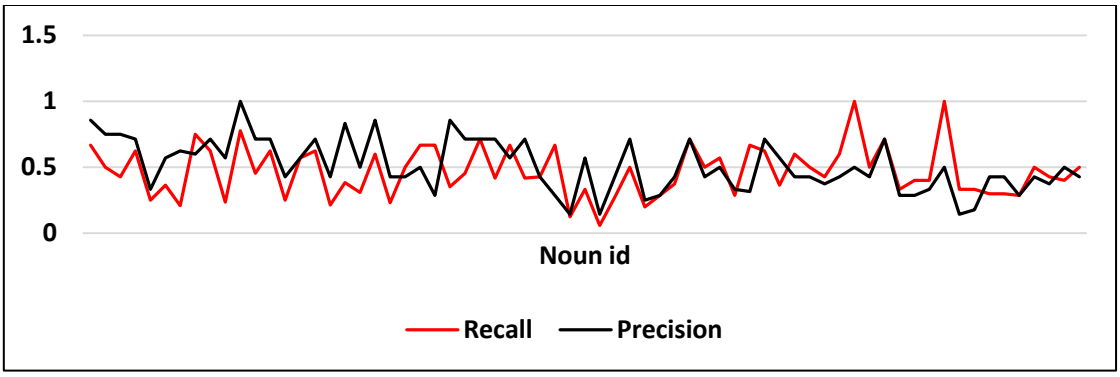


Figure 5.15 ROUGE Recall and Precision – Google Translate-based Dictionary

Figure 5.16 represents the average recall trend ($A(R|i)$), and Figure 5.17 represents the average precision trend (($A(R|i)$) at each run of the VSyn program (as introduced in section 4.2). Both figures show the stability of the recall and precision. In Almaany based evaluation or in Google Translate based evaluation, the recall and precision curves converge to their mean after process a few numbers of nouns (roughly after the noun number 40). This gives a good indication of the accuracy of the answer set generated by the VSyn program. If the recall and precision were very low in some cases and very high in other cases (large fluctuations) this requires the process of a large number of nouns to see the stability. The indication that can be obtained from the curves that in most cases, the precision and recall were close to their average. Also, it is important to note that the recall and precision were higher in the Google Translation based evaluation than in Almaany based evaluation.

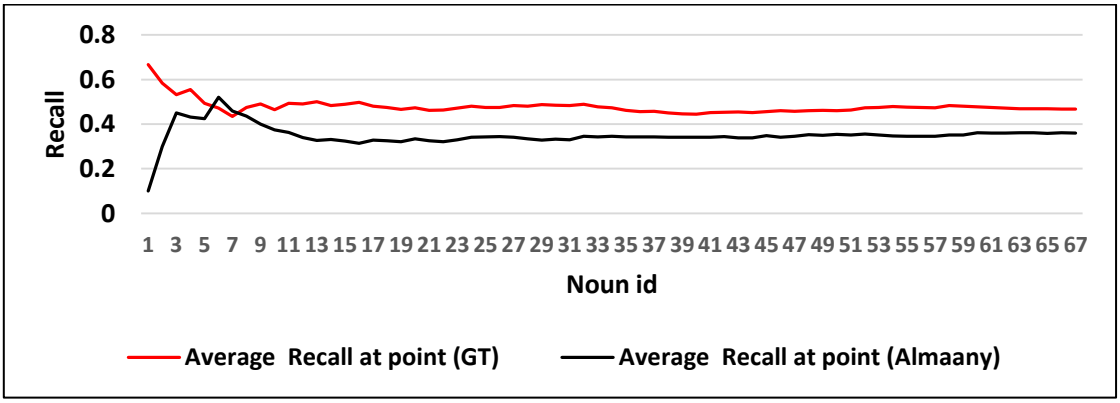


Figure 5.16 Average Recall Trends at each Noun Processed

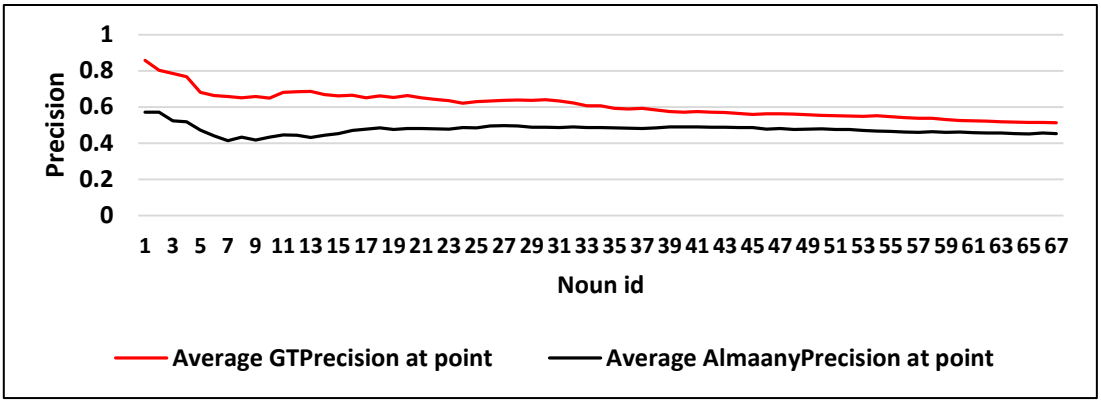


Figure 5.17 Average Precision Trends at each Noun Processed

5.2.2.3 Manual Evaluating using Arabic language experts

The recall and precision results revealed a significant performance of our system, but it showed that 53% of the synonyms are not returned, and the explanation was the nature of the base data used in the comparison. However, the satisfaction of the modern Arabic language native speakers about the accuracy of the synonyms sets returned by the NBDV method is necessary. The use of human evaluators used in many research publications (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), (Benabdallah, Abderrahim, & Abderrahim, 2017), and (Zhang, Li, & Wang, 2017). In our evaluation, two Arabic language experts and four Arabic language speakers from Prince Sattam bin Abdel Aziz University in Saudi Arabia voluntarily evaluated our system. They are divided into two groups; each group evaluated 82 nouns' synonyms. From their experience, the experts wrote the number of correct synonyms generated from our system for each noun. So the precision was redefined as follow:

$$\boldsymbol{Precision} = \frac{\boldsymbol{the\ number\ of\ automatically\ generated\ synonyms\ that\ the\ expert\ agreed\ their\ correctness}}{\boldsymbol{total\ number\ of\ automatically\ generated\ synonyms\ of\ the\ noun}}$$

Table 5.3 shows the evaluators' ratio of satisfaction (the precision value).

Table 5.3 the Average Precision for the Manual Evaluation

Group one of nouns containing 81 noun		Group two of nouns containing 83 noun	
Expert name	**P**	**Expert name**	**P**
Adeel(expert)	55%	Sadam(expert)	50%
Firas(speaker)	62%	Bassam(speaker)	57%
Sana(speaker)	60%	Nour(speaker)	61%
Average	59%	Average	56%
Average			57.5%

The Experts evaluation showed that the precision was significant; ranging from 50% to 62%, which proves the precision values gained in the Google Translate -based evaluation. Two of our experts Sadam and Adeel are Arabic Language specialists and they gave 55% and 50% of satisfaction, which represents a reasonable ratio. The other evaluators are Arabic language Native speakers, and their rate of satisfaction ranges from 57% to 62%.

Table 5.4 depicted that in Almaany based evaluation, the recall was somehow low, but the precision was significant. The recall reflects the fact that among all the synonyms found for a noun, 36% of them were returned, whereas the precision demonstrates the fact that among the synonyms returned by our system, 46% of them were relevant. Both the Google Translate based evaluation and the manual evaluation showed that more than half of the answer set elements generated automatically are correct synonyms, and the Google Translate based evaluation showed that around half of the synonyms found in Google Translate were retrieved by our method.

Table 5.4 Average Precision and Average Recall Using Three Evaluation Strategies

Evaluation Type	AR	AP
Dictionary Based Evaluation – Almaany Dictionary	36%	46%
Dictionary Based Evaluation - Google Translate Dictionary	47%	51%
Expert and Speakers Evaluation – Manual Evaluation	N/A	57.5%

5.2.3 Time Complexity Analysis

Assuming that the number of nouns in the whole corpus is N, the number of verbs in the entire corpus is v, and the number of all terms in the corpus is n. According to the NBDV method, to find the synonyms of the noun x, the necessary computational steps are listed in Table 5.5.

The worst-case time complexity in step 8 is O(N.v.n), this occurs if y = v and j = N, the meaning of y = v is that all the verbs in the dataset appeared with the noun x, and the meaning of the j = N is that all the nouns in the dataset are sharing the set of verbs stored in $S_{nv.}$ Actually, these conditions are impossible to happen because in real languages we cannot find one verb that comes with all the nouns or all the nouns share a specific set of verbs, so we can consider y, j as constant, which means the total complexity of the NBDV method will be O(j.y.n) + O(j.n) ⇒ O(n). Figure 5.18 supports our claim regarding the y possible value, the number of shared verbs processed by the OWS in each run of the NBDV system was counted, and the maximum obtained value of y is 829, the average value of y is 186 verb, and in 63% of the runs, the value of y was less than 200. Regarding j, the maximum value

of j recorded in our experiment was 521. Figure 5.19 shows that in 72% of the NBDV runs, the number of processed nouns is less than 100, and a very small ratio of runs processed a high number of nouns (10% of runs processed more than 200 nouns).

Table 5.5 Time Complexity Analysis of the NBDV model

	The NBDV operation	Expected time complexity	Description
OWS Computation	1) Extracting the set of verbs S_{nv} adjacent to the noun n, assume that the number of extracted verbs is y	O(n)	Scanning all the corpus elements with n number of terms.
	2) Computing each parameter mentioned in equations 14,16, and 17	O(n)	
	3) Computing the OWS weight	O(1)	
	4) Constrcting of the $\vec{x}$, Repeat step 2 and 4 for each verb $\in S_{nv}$	O(y.n)	In the worst case O(v.n) If v = y.
	5) Computing the Orbit range (layer in equation 19)	O(1)	
	6) Dividing the $\vec{x}$ vector components in sets according to the value of layer and create the s1,s2,s3.	O(y)	
	7) Take s1, s2, s3 and extract all the distinctive nouns x_i adjacent to verbs appeared in s1, s2, s3. Assume the number of extracted nouns is j	O(y.n)	In the worst-case O(v.n) If v = y.
	8) Repeating steps 2,3, and 4 for each noun extracted in 7, computing $\overrightarrow{x_i}$	O(j.y.n)	In the worst case O(N.v.n) If y = v and j = N
	Total OWS Complexity	O(j.y.n)	The max values of the complexity from step 1 - 8
Synonyms Detection	9) Computing sim $(\vec{x}, \overrightarrow{x_i})$, equation 20	O(j)	
	10) Extracting the synonyms, if sim $(\vec{x}, \overrightarrow{x_i}) > 18\%$	O(j.n)	In the worst case O(N.n) If j = N.

If the NBDV method hires the traditional tf.idf weighting scheme developed in CBoW and SG model (Mikolov, Chen, Corrado, & Dean, 2013), the computations required to compute the weight of each noun take O(n), and in step 8, this process is repeated n times because the CBoW and SG model compute the cosine similarity between x and all the terms found in the corpus, this implies that the time complexity of the CBoW and SG models is $O(n^2)$.

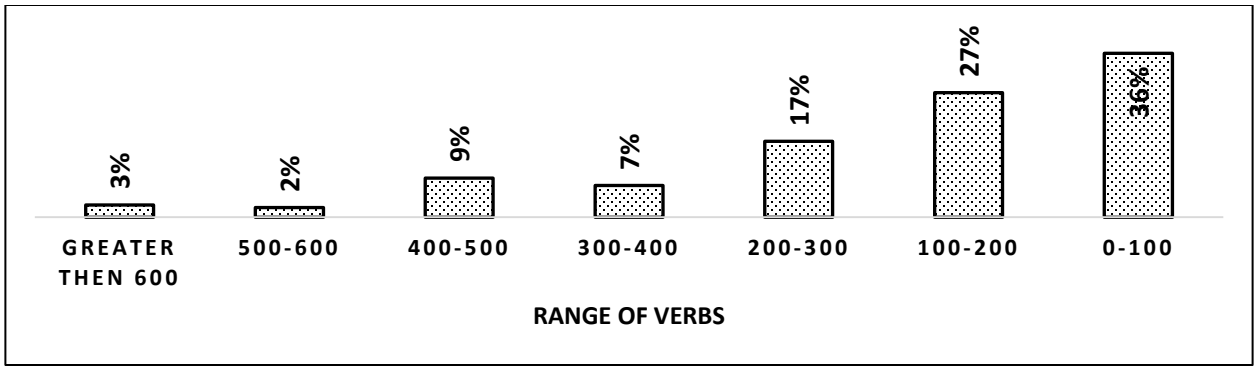


Figure 5.18 the Ratio of Processed Verbs in 564 Runs of the NBDV.

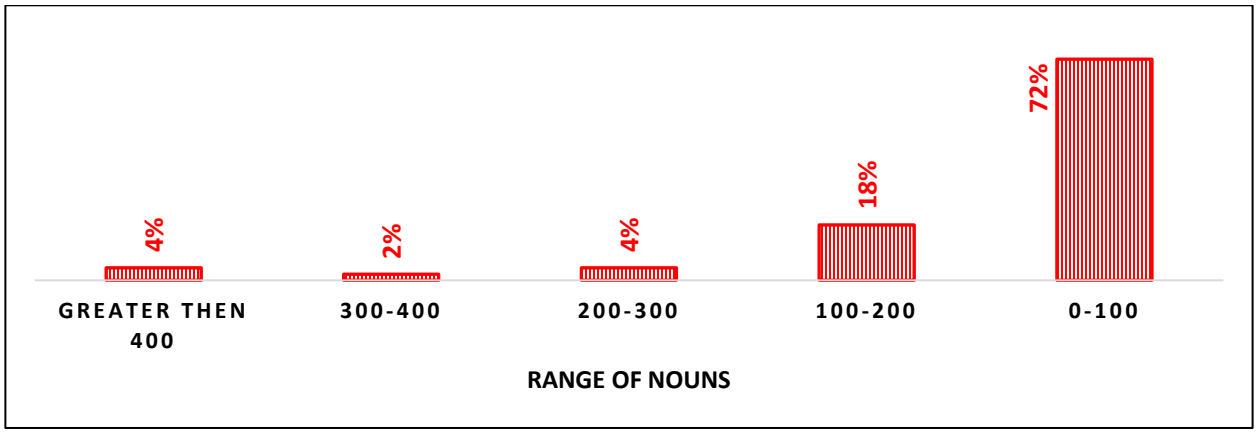


Figure 5.19 The Ratio of Processed Nouns in each Run of 564 Runs

5.3 Evaluation of Employing the MLS and NBDV in Information Retrieval System

The evaluation of employing the ATS semantic methods described in 3.3 and 3.4 on the relevancy and efficiency of the IR system that uses the VSM to match the documents and user query was done through the analysis of the results of the experiments Exp 1 to Exp 5. The Exp 1 – Exp 3 were performed without synonyms expansion, and those experiments test the employment of different ATS techniques in reducing the Inverted index and how this reduction affected the relevancy assessment. The Exp 4 and Exp 5 are performed to test the employment of different ATS techniques but with query expansion. The expansion is accomplished using the synonyms that are extracted by the NBDV method. The results are collected in section 4.3.2 that includes the relevancy results (recall, precision, Interpolated Average Precision, MAP) and the inverted index size. The size of the inverted index is used to measure the

enhancement achieved on the IR system performance. During the evaluation, we linked the recall-precision curve with the size of the inverted index and the final recall obtained at the end of each experiment.

In this section and all its subsections, the abbreviations appear in the figures have the following meanings: MC-curve refers to the recall-precision curve of the IR system that uses the main corpus to build the inverted index (without summarization), MLS-curve refers to the recall-precision curve of the IR system that uses the extracts corpus generated from the MLSExtractor, LSA-curve refers to the recall-precision curve of the IR system that uses the extracts corpus generated from the LSAExtractor, VSM-curve refers to the recall-precision curve of the IR system that uses the extracts corpus generated from the VSMExtractor, and JAC-curve refers to the recall-precision curve of the IR system that uses the extracts corpus generated from the JacExtractor.

5.3.1 The effect of the MLS on the IR relevancy results

This subsection assesses the relevancy results of Exp 1, 2, and 3. Figure 5.20 shows the recall-precision curves obtained in Exp 1. The curves trace the precision behavior at 11 recall points. The red curve represents the MC-based retrieval and the other curves represent the Extract based retrievals. In Figure 5.20, the red curve represents the optimal relevancy results generated from the IR system. Note the slight differences between the red curve and the other curves which mean that all the extracts based retrievals' results succeeded in retrieving a considerable number of relevant documents. The LSA-curve and MLS-curve show a small drop comparing with the JAC-curve and VSM-curve extracts retrieval, especially after r4.

The trend appears in Figure 5.20 should be supplemented with the other results obtained in Exp1; the inverted index size and the final recall and MAP. Figure 5.21 presents the final results of the recall, MAP, and the ratio of the extracts inverted index size relative to the main corpus inverted index. Note that the MLS-based retrieval obtained convergent MAP results with the other extracts based retrievals, and note that the recall value of the MLS-based retrieval was less than the MC-based retrieval by 12%. The recall value of the MLS-based retrieval is 65%, which represents a reasonable result because we obtained it at 58% reduction in the inverted index. The JAC-based retrieval and VSM-based retrieval relevancy results were very close to the MC-based retrieval results but with inconsiderable reductions in the inverted index size (21%, 32% respectively), and this comes compatible with the findings obtained in the evaluation of the VSMExtractor and JacExtractor in section 5.2.

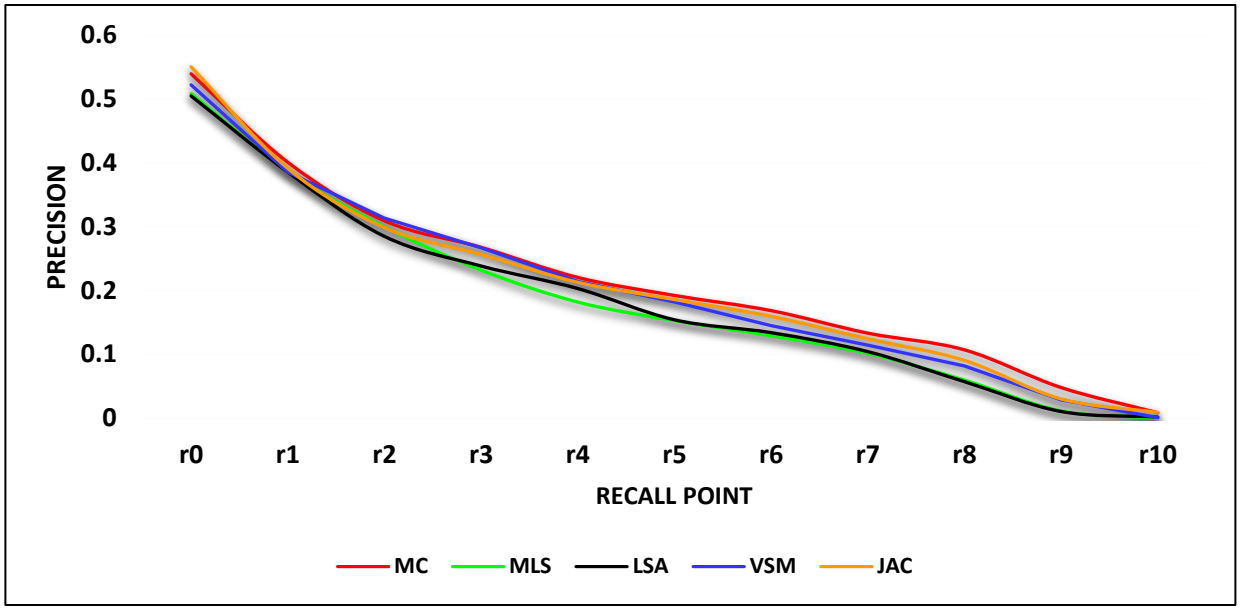


Figure 5.20 the Recall-Precision Curves in Exp1

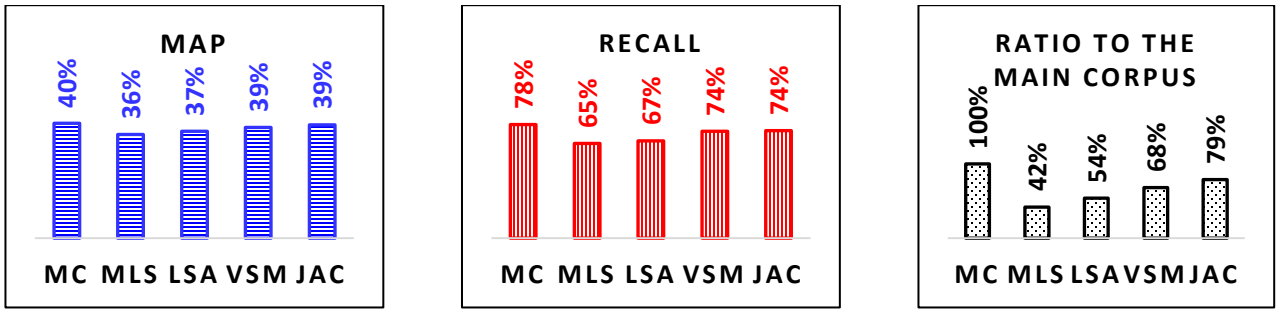


Figure 5.21 MAP, Recall, and the Ratio to the Main Corpus Size (in Exp 1).

Comparing with Exp1, two changes were made In Exp 2: 1) the number of queries was increased to 100, and the purpose was to test the behavior of the IR system with a larger number of queries, 2) the answer set retrieved from the MC-based retrieval was taken as the relevant set of documents. The inverted indexes in Exp1 and Exp2 are the same. Figure 5.22 shows the recall-precision curves obtained from the Exp 2.

Note that the MC red line always 1 because we considered it as the gold answer. From r0 to r5, all the extracts based retrievals' relevancy results were very close to the MC-based retrieval relevancy results, after r6, the extracts based retrievals starts to drop and the largest drop happened to the MLS-based retrieval. The trend appears in Figure 5.22 should be supplemented with the other results obtained in Exp2; the inverted index size and the final recall and MAP.

Figure 5.23 presents the final results of the recall, MAP, and the ratio of the extracts inverted index size relative to the main corpus inverted index. Note that the MAP was very high for the four extracts based retrieval and this reflects that the extract-based retrievals obtained reasonable precision. The recall behavior is the same as the one that appeared in Figure 5.21, the MLS-based retrieval obtained 63% recall value at 58% inverted index reduction, the recall of the VSM-based retrieval and JAC- based retrieval was above 80% but the amount of reduction was inconsiderable. The relevancy assessment of Exp1 and Exp 2 are roughly the same, the extracts based retrieval curves are very close to the curve obtained in the MC retrieval, and the MAP and recall have the same behavior.

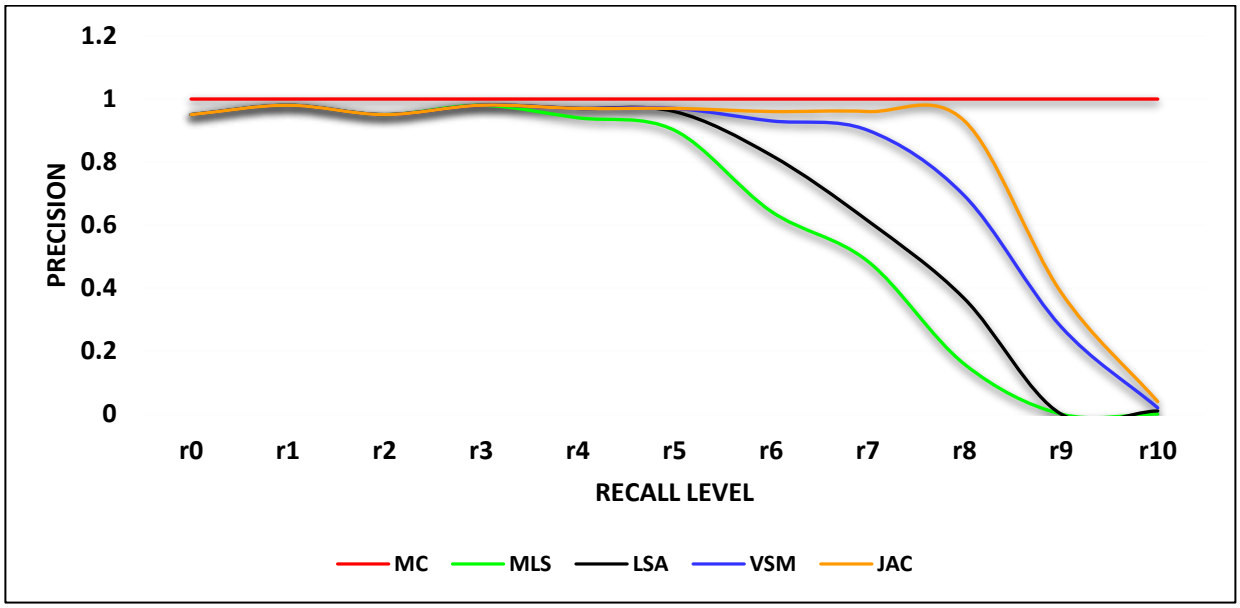


Figure 22 the Recall-Precision Curves in Exp2

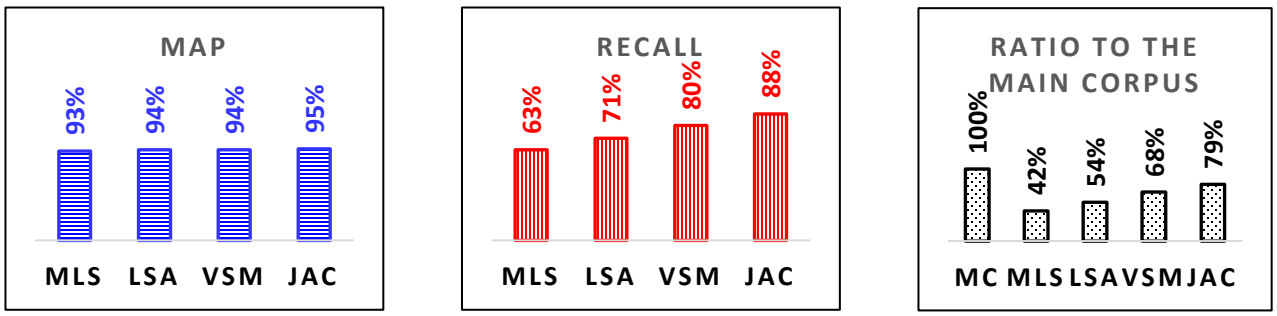


Figure 5.23 MAP, Recall, and the Ratio to the Main Corpus Size (in Exp 2).

In Exp 3, an English language corpus was used instead of the Arabic Language corpus, as described in chapter 4. The purpose of performing the Exp 3 is to measure the effect of each extractor on the relevancy of the IR system when the corpus is not semantically rich (the text's writers do not diversify their vocabularies).

Figure 5.24 shows the recall-precision curves obtained in Exp 3. From r0 to r4 all the extracts based retrievals relevancy results were very close to the MC-based retrieval relevancy results, after r5 the extracts based retrievals starts to drop and the largest drop happened to the MLS-based retrieval.

In this experiment, the VSM-based retrieval obtained fewer precision values than the LSA-based retrieval because the used corpus contains the posts of young people bloggers who normally do not diversify their vocabularies during the posting. This feature in the English corpus magnified the role of the second layer in the MLS extraction (VSM layer) and caused the VSM extraction to delete a large portion of the text based on simple statistical calculations (the role of semantic analysis is weak in this case). Also, this feature affected the precision values of the MLS-based retrieval because the MLS extractor uses the VSM model in the middle layer, as described in section 3.3. The drop in the VSM-curve caused the drop of the MLS-curve, and it was not a drop that is based on the semantic analysis of the text.

In Figure 5.25, we see that the recall of the VSM-based retrieval and MLS-based retrieval is 54% and 55%, respectively. Also, the MLS-based retrieval and VSM-based retrieval showed 39% and 36% reduction in the inverted index size, which is not the desirable ratio. The obtained results in Exp 3 proves one of our claims that the simple statistical analysis of the text analysis based on the term frequency and term distribution hurts the extraction results, and this affected the relevancy of the IR system that uses an inverted index that is reduced by the VSM extractors such as the IR systems appeared in (Perea-Ortega J. M.-L., 2013), (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001).

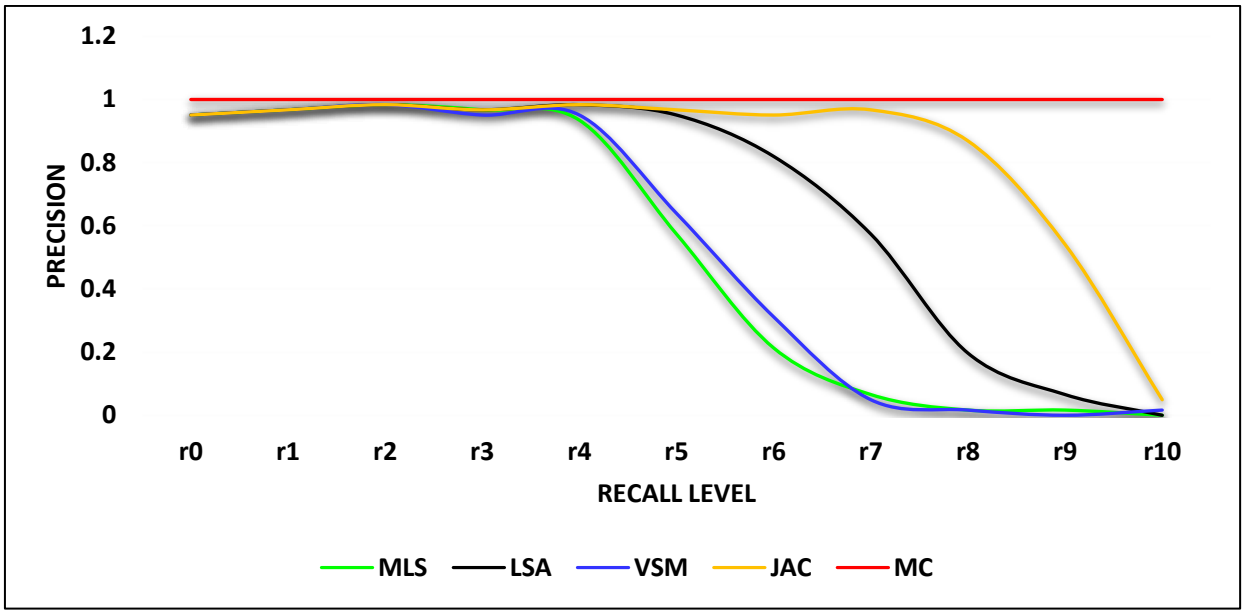


Figure 5.24 the Recall-Precision Curves in Exp3

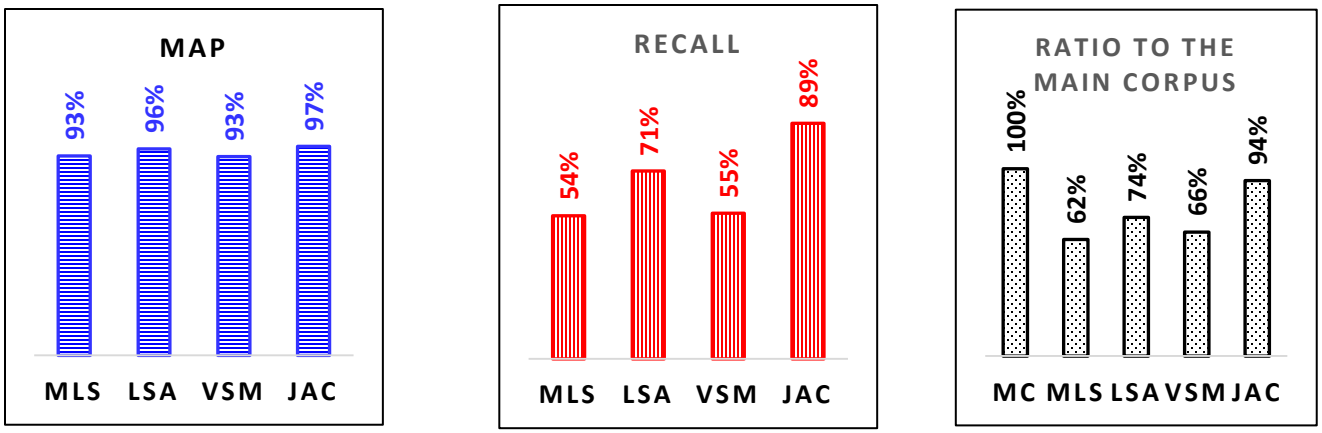


Figure 5.25 MAP, Recall, and the Ratio to the Main Corpus Size (in Exp 3).

5.3.2 The Effect of the MLS with NBDV on the IR system

Exp 4 and Exp 5 use the same experimental settings used in Exp1 and Exp 2 respectively, but in Exp 4 and Exp 5, we expanded the user query with the synonyms generated from the VSyn system that was developed based on the NBDV synonyms extraction method. As described in section 4.3, the same results collected in Exp 1 and Exp 2 were collected in Exp 4 and Exp 5, so we can make a comparison between the relevancy measures with and without expansion. Note that the text extraction was already accomplished and the extracts inverted indexes were prepared in Exp1 and Exp2. The purpose of Exp 4 and Exp 5 was to investigate the enhancements on the relevancy after

the NBDV expansion. Figure 5.26 shows the relevancy results of the MLS-based retrieval with NBDV synonyms expansion (in Exp 4 and 5) and without NBDV synonyms expansion (in Exp 1 and Exp 2). Note that we have a small improvement in the recall in both experiments (1%), and the precision in Exp 4 was increased by 1% and remained stable in Exp 5 at 93%. Even if these improvements are simple, they are important because in the Exp 1 the MAP of the MC-based retrieval was 40% (as shown in Figure 5.21) and the synonyms expansion makes the MAP of the MLS-based retrieval more close to the MAP of the MC-based retrieval. In Exp 5 the MAP is already high (93%), so the expected improvement in the MAP is small. Regarding the recall, in both experiments, the achieved enhancement was shy, only 1%. The important note that appeared in Exp 4 and Exp 5 is that no relevancy measures were hurt by the expansion, which gives a strong indication about the accuracy of the answer set retrieved from the NBDV method of synonyms extraction. This note is supported by the recall-precision curves obtained in Exp 1 and 4, and in Exp 2 and 5 as shown in Figure 5.27. The curves on the left side and on the right side of Figure 5.27 are roughly identical, so in the worst case, if the expansion did not give the desired improvement, it did not hurt the precision or recall.

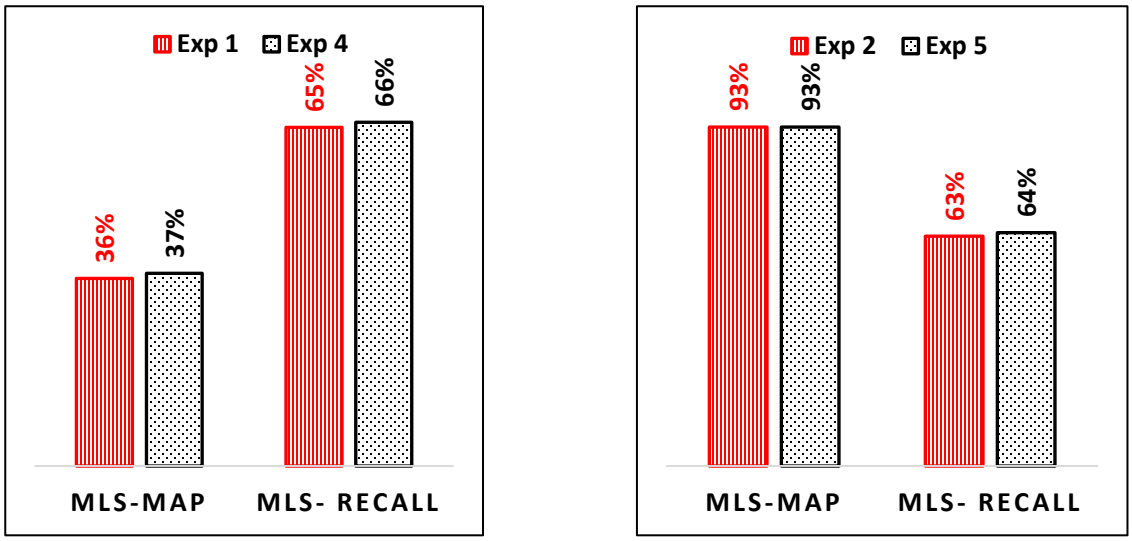


Figure 5.26 the Relevancy Results of the MLS-based retrieval with and without NBDV Synonyms Expansion

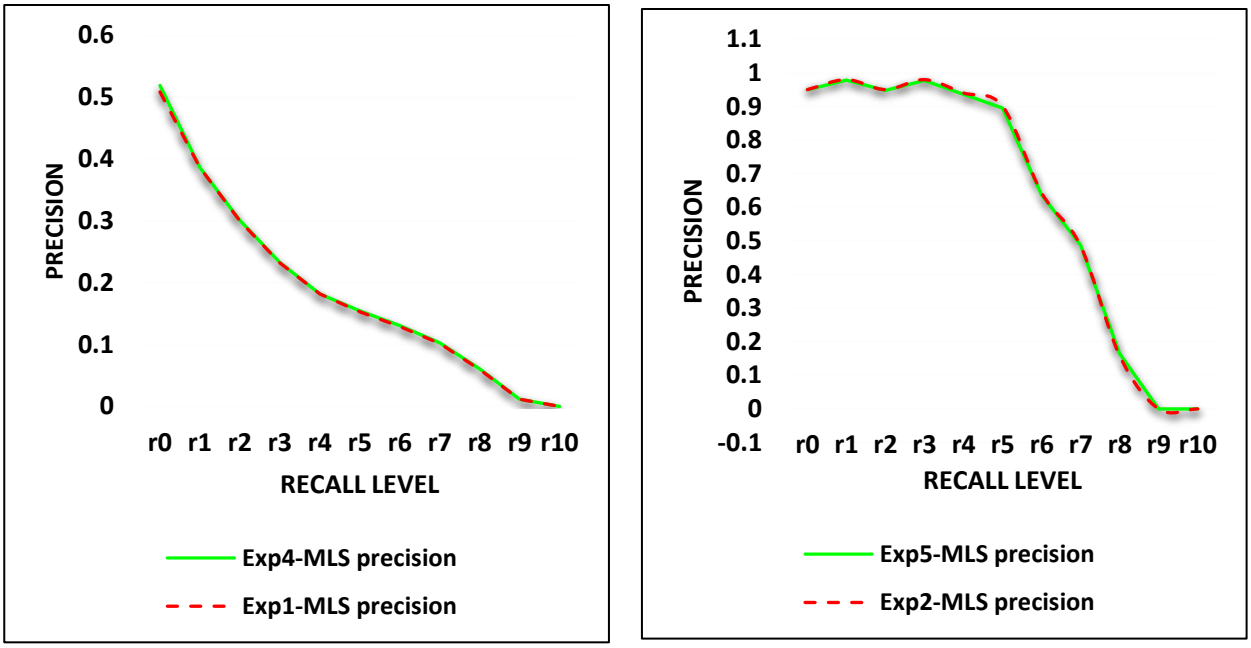


Figure 5.27 the Recall-Precision Curve in Exp 1 and 4 (Left Side), and in Exp 2 and 5 (Right Side).

5.4 Discussion

This section discusses the advantages and drawbacks of the MLS and NBDV extraction methods as methods of extraction and as a tool that can support the IR systems. Also, comparisons with other research in the field are established to show the enhancement and improvement achieved.

5.4.1 MLS Text Extraction Discussion

The discussion of the MLS text extraction considers two perspectives, the proposed extraction procedure and the new evaluation process that has been used to assess the quality of the automatically generated extracts. We introduced a new method for text extraction based on the sentences' resemblance, and we developed a new evaluation technique that can give us a clear picture of the automatic extracts actual contents. Table 5.6 presents a sample of the automatically generated extracts[17].

From our results and evaluation, the technique of extraction has the following advantages:

[17] Parts of this section and its subsections are mentioned in the second paper in the "**Publications Arising from This Thesis**" section

No based sentence - such as the user recommendation or the document title- was used as a base of extraction: The extraction process performs the recursive computation of the similarities between the sentences found in the document. The similarities values reflect the degree of the verbatim, statistical, and semantic resemblance. For example, the similarity values between the sentence number 4 and the sentence number 11 in document 1 exceeded the threshold value in all the automatic extraction systems developed in this work (57%, 99%, 90%, and 57%). And if we consider the meaning of the two sentences, we find that both of them are talking about the appearance of the Beethoven talent in the music from an early age. The sentences are:

قدم أول عمل موسيقي في سن 8 سنوات

He presented the first musical work at the age of 8 years.

ظهر تميزه الموسيقي منذ صغره، فنشرت أولى أعماله وهو في الثانية عشرمن عمره سنه 1783 م.

His musical excellence appeared from a young age; he published the first musical work when he was twelve-year-old in 1783.

Table 5.6 the Generated Automatic Extract for Document 2 in Essex Corpus

VSMExtractor Extract	ذكر تقرير إخباري أول من أمس أن شهر مايو المشمس يشهد أكبر عدد من حالات الانتحار. وقال باحثون بريطانيون إن عدد حالات الانتحار يزيد في شهر مايو المشمس ليكون أكثر من أي شهر آخر وهم يعتقدون أن الأمر راجع إلى حالة الطقس. ويقول المسؤول عن الخدمات الصحية في المجموعة البروفيسور كريس تومسون إن هناك علاقة مباشرة بين سطوع الشمس والانتحار. وتبين الإحصائيات أن عدد محاولات الانتحار ارتفع بنسبة 50 % منذ 1990 وأن معظم من أقدموا على الانتحار كانوا من الرجال. News report reported yesterday that the sunny May has the highest number of suicides. British researchers said the number of suicides increased in May to more than any other month and they thought it was due to the weather. Professor Chris Thompson, the group's health services officer, says there is a direct relationship between sun brightness and suicide. Statistics show that the number of suicide attempts has increased by 50% since 1990 and that most of those who committed suicide were men.
JacExtractor Extract	ذكر تقرير إخباري أول من أمس أن شهر مايو المشمس يشهد أكبر عدد من حالات الانتحار. وتقول مجموعة برايوري المتخصصة في بحوث الطب النفسي إن الطقس المشمس الذي عادة ما يساعد الناس في التغلب على كآبتهم يعطيهم كذلك القدرة على اتباع دوافعهم الانتحارية. اوضحت دراسات أخرى أن مستوى السيروتونين يرتفع حسب كمية أشعة الشمس التي يتعرض لها الشخص. وتبين الإحصائيات أن عدد محاولات الانتحار ارتفع بنسبة 50 % منذ 1990 وأن معظم من أقدموا على الانتحار كانوا من الرجال. News report reported yesterday that the sunny May has the highest number of suicides. Brownie, a specialist in psychiatric research, says sunny weather, which often helps people overcome their depression, also gives them the ability to follow their suicidal motive .Other studies have shown that the level of serotonin increases according to the amount of sunlight the person is exposed to receive. Statistics show that the number of suicide attempts has increased by 50% since 1990 and that most of those who committed suicide were men.
LSAExtractor extract	LSAExtractor extract ذكر تقرير إخباري أول من أمس أن شهر مايو المشمس يشهد أكبر عدد من حالات الانتحار. وتقول مجموعة برايوري المتخصصة في بحوث الطب النفسي إن الطقس المشمس الذي عادة ما يساعد الناس في التغلب على كآبتهم يعطيهم كذلك القدرة على اتباع دوافعهم الانتحارية. وتبين الإحصائيات أن عدد محاولات الانتحار ارتفع بنسبة 50 % منذ 1990 وأن معظم من أقدموا على الانتحار كانوا من الرجال. News report reported yesterday that the sunny May has the highest number of suicides. Brownie, a specialist in psychiatric research, says sunny weather, which often helps people overcome their depression, also gives them the ability to follow their suicidal motive .Statistics show that the number of suicide attempts has increased by 50% since 1990 and that most of those who committed suicide were men.
MLSExtractor extract	ذكر تقرير إخباري أول من أمس أن شهر مايو المشمس يشهد أكبر عدد من حالات الانتحار. قال باحثون بريطانيون إن عدد حالات الانتحار يزيد في شهر مايو المشمس ليكون أكثر من أي شهر آخر وهم يعتقدون أن الأمر راجع إلى حالة الطقس. ويقول المسؤول عن الخدمات الصحية في المجموعة البروفيسور كريس تومسون إن هناك علاقة مباشرة بين سطوع الشمس والانتحار. وأوضحت دراسات أخرى أن مستوى السيروتونين يرتفع حسب كمية أشعة الشمس التي يتعرض لها الشخص. News report reported yesterday that the sunny May has the highest number of suicides British researchers said the number of suicides increased in May to more than any other month and they thought it was due to the weather. Professor Chris Thompson, the group's health services officer, says there is a direct relationship between sun brightness and suicide. Other studies have shown that the level of serotonin increases according to the amount of sunlight the person is exposed to receive.

Language Independent: No linguistics features were considered during the extraction process, all the similarity calculations were statistical, and they are applicable to another language.

Domain-Independent: Unlike the features-based extraction and domain-based extraction proposed by Nekota and McKeown (Nenkova & McKeown, A survey of text summarization techniques, 2012), no particular domain features affected our technique of extraction.

The deletion procedure used in the extraction systems was robust; it deletes the repeated sentences based on a well-defined process and parameters. It discards 58% of the text and obtained reasonable levels of HIGHC and FULLC containment (27%) (Figure 5.1 - 5.4). Also from Figure 5.6 and 5.7, the ROUGE results showed reasonable recall value (48%) which was higher than the recall values achieved by the researchers in (Babar & Patil, 2015) and (Chen, et al., 2015) who used the LSA technique in their summarization systems. The MLSExtractor gave 41% f-score values which were higher than the f-scores value obtained in (Yeh, Hao-RenKe, Yanga, & Meng, 2005), (Mashechkin, Petrovskiy, Popov, & Tsarev, 2011), and (Chen, et al., 2015). The precision values of the VSMExtractor surpassed the precision value achieved by Kiyoumarsi in (Kiyoumarsi, 2015), who used the VSM to summarize text documents. Figure 5.28 compares the recall value obtained by MLSExtractor and LSAExtractor with Kiyoumarsi extractor (Kiyoumarsi, 2015), Babar and Patil extractor (Babar & Patil, 2015), with Chen Extractor (Chen, et al., 2015), Yousefi and Hamey (Yousefi & Hamey, 2017), and Tayel Extractor (Tayal, Raghuwanshi, & Malik, 2017). These extraction systems were developed recently (2015-2017), and the authors used statistical techniques and employed the ROUGE tool to evaluate their summaries. Figure 5.28, clearly shows that both MLSExtractor and LSAExtractor achieved higher recall than the other systems.

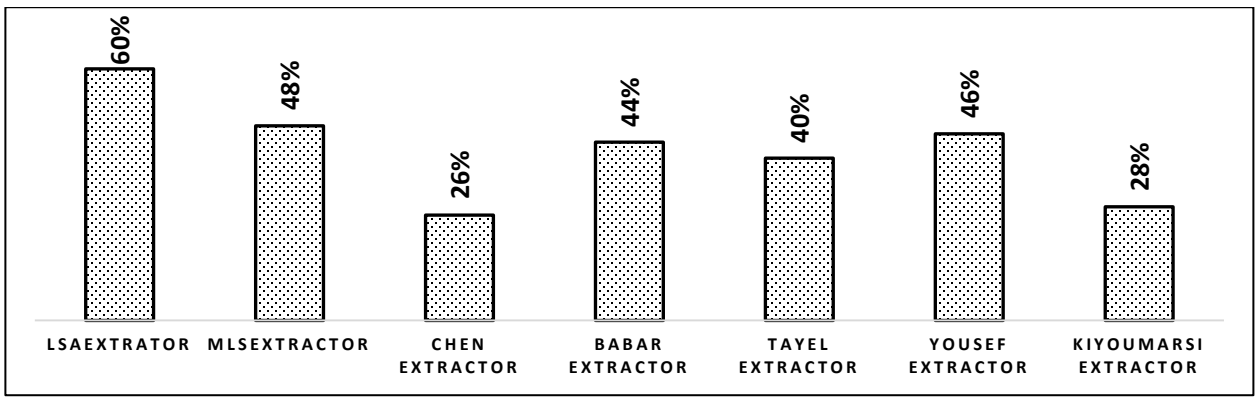


Figure 5.28 Comparison of the Recall Values between MLSExtractor and LSAExtractor with Recent Extractors.

The use of different approaches for similarity calculation allows us to measure the output of our deletion process at three levels of complexity: simple statistical (Jaccard coefficient), complicated statistical (VSM), and complex statistical with semantic analysis (LSA, MLS). From Figure 5.3, 5.4, 5.6, and 5.7, we found that the semantic investigation of the text obtained the most significant containment values (LOWC containment less than 26%) and ROUGE values (R 60% and P 41%). Both the MLSExtractor and the LSAExtractor work in the semantic level and both of them obtained convergent ROUGE and Containment evaluation results, but the MLS extraction surpassed the LSA Extraction in the condensation rate, and it succeeded to reduce the size of the text to 42% instead of 54% achieved by the LSAExtractor. The summary of the MLSExtractor evaluation results can be drawn from Figures 5.4, 5.5, 5.6, and 5.7 the CR was 42%, the Average recall was 48%, the average precision 40%, with 48% MODC Containment, 16% HIGHC Containment, and 2% FULLC Containment.

The containment evaluation approach proposed in this work provides an accurate judgment about the contents of the automatic extracts: It measures the percent of the sentences from the reference extract that appeared in the automatic extract and takes the size of the automatic extract into consideration. The AR and AP generated from the ROUGE 2.0 tool give a general indication of the extraction quality, but with variable sized automatic extract, the ROUGE gives misleading assessment because it measures extracts of different sizes. ROUGE evaluation provides an indication of the quality of the extracts, but it cannot judge accurately the percent of complete sentences that are shared between the automatic and reference extracts, and it does not consider the size of the extract. The Containment evaluation combines the RSI with the CR in one evaluation scheme. Figure 5.1 – 5.5 showed that the MODC, HIGHC, and FULLC Containment for the JacExtractor was 97%, for the VSMExtractor was 81%, for the LSAExtractor was 74%, and for the MLSExtractor was 66%. If we combine these results with the CR values and implement them in pairs (Containment, CR), we will get the pairs (97%, 79%), (81%, 68%), (74%, 54%), and (66%, 42%) for the JacExtractor, VSMExtractor, LSAExtractor, and MLSExtractor respectively. These pairs are important because they showed the percent of the sentences from the reference extracts found in the automatic extracts to the percent of text size reduction. For example, in JacExtractor, 97% of the references extracts sentences found in the automatic extracts at a condensation rate of 79% (only 21% from the text removed). These pairs clearly show that the MLSExtractor yielded the most acceptable results by matching the percent of correct retrieved sentences (66%) to the automatic extract size (42% of the original text).

Performance increase: the time complexity analysis accomplished in this paper showed that the complexity of LSA extraction is high comparing with VSMExtractor and JacExtractor. Thus we proposed the MLS extraction that reduced the number of runs of the LSA procedure and reduced the size of the original matrix. The MLS extraction calls the LSA procedure if the Jaccard Similarity and the cosine similarity was less than 50%. From Figure 5.10, the number of executions of the LSA procedure is 23500 in LSAExtractor and 11438 in MLSExtractor. The number of runs of the LSA procedure in the MLSExtractor is less than the number of runs in classical LSA extraction by 52%. Furthermore, in MLSExtractor, the SVD needs to run for only 35% of the terms and sentences found in the original matrix (especially for large documents, see Figure 5.11). This is a significant result because it increases the acceptability of employing the LSA in text mining.

The use of variable size Condensation Rate: this feature allows us to create a condensed version of the document that contains all the salient parts of the document and helped us to develop an accurate evaluation. Firstly, only the condensation rate was able to show that the MLSExtractor surpassed the other automatic extraction system. Secondly, if we used fixed CR, we cannot pretend that one extractor- from the three extraction systems developed in this work - performed better than the others because, in some cases, the size of the extract prevents the system from including more sentences, which may create the difference.

Also, a comparison with existing automatic methods was established. The comparison showed the bright side of our extraction method. Besides the important recall and precision values obtained, the system showed stable behavior, and in most cases, it returns recall and precision values that were close to their mean. Figure 5.8 explained that the values of recall, precision, and f-score were the highest compared with the obtained ROUGE results by UTF-8 SUPPORT and API tool extraction systems. Also, The MLS Extractor obtained 9% standard deviation for recall values and 14% for precision values. Comparing that with the existing methods, UTF-8 SUPPORT TOOL and API, it is found that our MLS Extraction system obtained lower standard deviation than the standard deviation obtained from the UTF-8 SUPPORT TOOL and API systems (see Figure 5.11.a and 5.11.b)

5.4.2 NBDV synonyms Extraction Discussion

In this research, an efficient statistical method is developed to extract synonyms, and the time complexity analysis showed that the method needs O(n) to extract the synonyms of a specific noun. However, an accuracy comparison

of our method with other publications in the field is necessary. Figure 5.29 presents a comparison between the results that are mentioned in Figures 5.14 – 5.17 and Table 5.3 with other research in this field. In (Henriksson, Moen, Skeppstedt, Daudaravicius, & Duneld, 2014), (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), (Lonneke & Jorg, 2006), and (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012) the precision calculations are presented in a clear manner. The authors of these references experienced different approaches of synonyms extraction (Henriksson used enhanced distributional hypothesis model, Leeuwenberga used Statistical approaches with relative cosine similarity, Lonneke used machine translation approach, Lobanova used learning approach, and Minkov used graph-based approach)

Figure 5.29 reveals that the precision of the NBDV was significant compared with other statistical methods used for synonyms extraction, (see Figure 5.17 and Table 5.3 for detailed results) the system obtained 51% average precision in the dictionary-based evaluation in which the synonyms generated by VSyn system were matched against the synonyms taken from online dictionaries. This precision value was less than the precision obtained by Minkov and Cohen (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012) by 8% and greater than the precision of the other systems. Minkov and Cohen (Minkov & Cohen, Graph based similarity measures for synonym extraction from parsed text, 2012) used a path constrained graph, and the problem with this graph is the time required to construct the graph and the space needed to store the graph. The graph stores each term in the corpus with all existing edges that link this term to the other terms found in the corpus. Add to this, the time needed to follow all the paths that lead to the terms. So, the improvement in the time obtained in the NBDV method is much more important than the 8% loss of accuracy, especially that the precision of the NBDV method was more than 50%. Table 5.5 and Figure 5.18, 5.19, depicted the time analysis of the NBDV method and showed the improvement in the synonyms extraction efficiency. To be more accurate, the average number of verbs processed was 186 verbs for each noun, and the maximum number of verbs processed was 839 for the noun "عمل work". So in our method, the determination of the semantic relations between a specific noun and the other nouns found in the corpus is performed by processing (weighting) 186 verbs appeared with that noun. Comparing with the CBoW, SG, and relative cosine similarity (Leeuwenberga, Vela, Dehdar, & Genabith, 2016), the improvement in the time consuming came in the (1) terms weighting step and in the (2) similarity computations between the nouns.

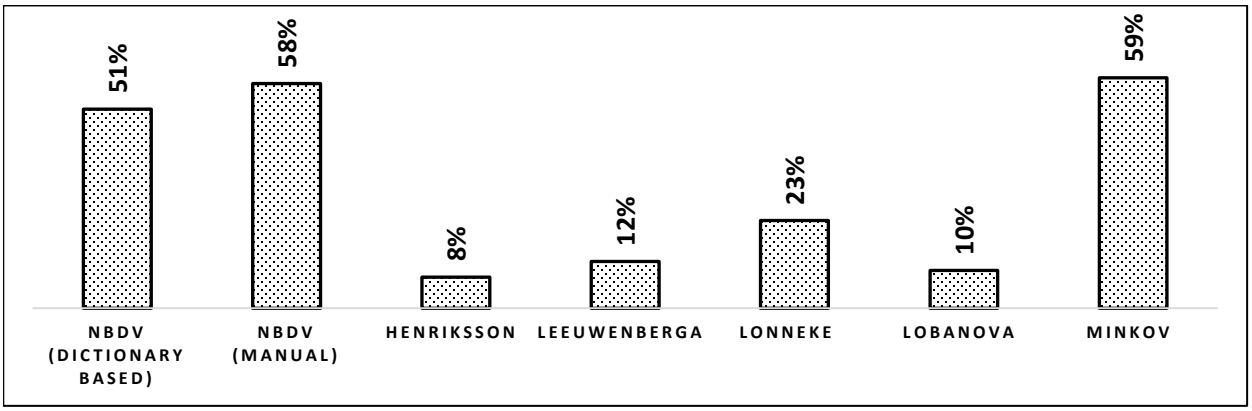


Figure 5.29 Precision Comparison with Existing Synonyms Extraction Systems.

In (Benabdallah, Abderrahim, & Abderrahim, 2017) (Zhang, Li, & Wang, 2017), the manual evaluation yields 76%, 80% average precision, respectively. The manual evaluation depends on the assessors' point of view and knowledge. In the manual evaluation accomplished in this research, we notice a difference between the experts' and natives' decision in whether the words are related words (hyponym, hypernym, plural) or synonyms. For example, for the noun "type نوع" the system returned "نوع، نمط، طراز، صنف، فِرق، رمز " as synonyms, Sana wrote five as correct synonyms, excluded "رمز symbol" whereas Adeel wrote four excluded "رمز symbol" and "فِرق sects". Adeel considered "فِرق sects" as a hyponym of the word "type نوع". Another difference between the speakers' judgment and the expert judgment was the plural, the speakers considered the plural as synonyms, for example, that VSyn generates the following synonyms for the word "area":

منطقه Area	منطقه Area	مدينه City	شباب youth	ولايه State	ولايات States	مناطق Areas	عديد numerous

Nour considered "areas" as synonymous to "area", whereas, Sadam excluded it. Sadam also excluded the word "states".

The factors that affected the precision are summaries in three factors: the first one was the existence of a set of nouns that are not synonyms (sometimes antonyms) and shared a set of distinctive verbs. For example, the direction names (north, south, east, and west), the month names, and the currency names. To see the effect of such kind of nouns, consider the synonyms set generated for the noun " جنوب South", the VSyn system produced the following candidate synonyms:

جنوب	شمال								شماليه							شرق
south	north	…	…	…	…	…	…	…	north	…	…	…	…	…	…	east

The nouns represent the four directions, and they are not synonymous with each other, but the question is why the system returned them. To answer this question, we returned to our results and scanned the verbs appeared with the noun *south,* see the following samples from our corpus:

...تهب الرياح شمالا.	... The wind blows to the north.
...شق طريقة الى الشمال.	... Cleave its road to the north.
...يمتد من شمال الكوكب.	... Extend from the north of the planet.
...تقع شمال الصين...	... located in the north of China ...
...يبعد نحو 77 كيلو متر الى الشمال ...	... about 77 km to the north ...
...يبدو في شمال الكرة الارضية..	... appear in the North of the globe...
...تتجه نحو الشمال...	... heading to the north ...
..تظهر شمال...	..appeared in the north ...
...التي كانت تتمكز شمالا.	... which was stationed north.
...تبدأ قافلة الشمال الاول بدخول...	... the first north convoy starts to enter ...

The replacement of the word north with south, east, or west in these sentences is acceptable, and lexically, the sentences remain correct. This means that the verbs (underlined above) are common for the nouns (north, south, east, west) and they have a convergent effect on the weighting computation. Therefore, the NBDV method considered north, south, east, and west as synonyms. The verb "تتجه head" appeared with these four nouns and got significant weights. The weight of this verb with respect to the north word was 32%, with respect to south 28%, with respect to east 29%, and with respect to west 27%. These weights and the weights of other common verbs produced high similarity between the nouns north, south, east, and west.

The second factor degraded the precision in the case of the dictionary-based evaluation was the synonyms set of the base dictionary; in some cases, it was empty. See the following samples:

Id	Noun	Almaany Set of synonyms	The Automatically generated synonyms
N6	دينار	Not Mentioned	يورو،سنه،طريق،وان،ريال،طاقه,,دينار
N17	شركه	Not Mentioned	شركه، شركات، وزاره، منظمه، مؤسسه، عام، مشاركه
N204	بيانات	Not Mentioned	بيانات، معلومات، نتائج، طلاب، كافه، نسبه، مصادر

The nouns " دينار Dinar ", "بيانات data", "شركة corporation" did not have synonyms in Almaany dictionary, In such case, the precision computation is not possible because no base set of synonyms is found, so we neglected them during the precision computation. This distorts the results (appeared in Figure 5.14 and 5.15) because the precision for the same nouns in the manual evaluation showed reasonable values. See the next sample:

Noun	The Automatically generated synonyms	Adeel Evaluation		Firas Evaluation		Saddam Evaluation	
		NUMBER	P	NUMBER	P	NUMBER	P
N6	يورو،سنه،طريق،وان،ريال،طاقه,,دينار ,	3	43%	3	43%	3	43%
N17	شركه، شركات، وزاره، منظمه، مؤسسه، عام، مشاركه	4	57%	6	86%	5	71%
N204	بيانات، معلومات، نتائج، طلاب، كافه، نسبه، مصادر	4	57%	5	71%	4	57%

The third factor that also affected our precision value was the mistakes found in the part of speech tagging produced in Kalimat dataset, for example, the candidate synonyms set produced by our systems for the noun " الدستور" contains the following: {مشروع,ثلاثه ، كلمه ، ايضا ، دستور ، قانون ، قرار } , the word "ايضا" is a Stopword means "*also"* or "*as well"*. The mistake in this case directly affected the precision because it happened in the synonyms set. Also, some tagging mistakes affected the weighting phase of our system, for example, the verbs list of the noun "حروب" contains the noun "دخان" smoke.

5.4.3 Advantages and Disadvantages of employing the MLS and NBDV in information retrieval.

The advantages of employing the MLS and NBDV extraction on the IR system developed in this research can be summarized in the following:

Condense version of the main corpus inverted index. The inverted index of the MLS in the Exp 1, 2, 4, 5 was less than the inverted index of the main corpus by 58%, and by 38% in Exp 3. Note that the sizes of the inverted indexes in the VSM and JAC-based retrievals were large and close to the inverted index size in the MC-based retrieval.

Convergent relevancy results between the MLS-based retrieval and the MC-based retrieval. As shown in the figures of section 5.4, the MAP and the recall-precision curve of the MLS-based retrieval were very close to their counterparts in the MC-based retrieval. In Exp 1, the MAP of the MC-based retrieval was 40% and the MAP of the MLS-based retrieval was 36%, and the latter is improved to 37% with NBDV expansion, as shown in Figure 5.26. In Exp 2, the MAP of the MLS-based retrieval obtained 93% of the MAP of the MC-based, as shown in Figure 5.23. The recall-precision curves of the in Figures 5.20, 5.22, 5.24 are very close until r5.

Convergent relevancy results between the MLS-based retrieval and the LSA-based retrieval. As shown in figures 5.20, 5.21, 5.22, 5.23, the recall, MAP, and the recall-precision curve of the MLS-based retrieval were very close to their

counterparts in the LSA-based retrieval. These relevancy results are obtained at 42% CR in MLS-based retrieval and at 54% CR in LSA-based retrieval, as appeared in Figures 5.6. Also, these results should be linked to Figures 5.10, 5.11 and Table 5.2 that showed the efficiency enhancement of the MLS extraction over the LSA extraction. The relevancy results in the MLS-based retrieval were achieved with 58% reduction in the number of executions of the LSA function, and the SVD manipulated 35% of the text.

From our results and evaluation, the following drawbacks were found which affected the IR relevancy results:

The role of each layer in the MLS extraction is corpus dependent. The part of the text that should be processed in each layer depends on the diversity of the vocabularies used to build the corpus. In Exp 3, because the vocabularies do not contain the required diversity, the second layer (VSM layer) had the greatest effect in the MLS extraction, and this caused to decrease in the role of the semantic layer (upper layer). In Figures 5.24 and 5.25 that showed the results of Exp 3, the MLS-based retrieval and the VSM-based retrieval had convergent relevancy results which disagree the other experiments that showed convergent results between the MLS-based retrieval and LSA-based retrieval extraction (in Figures 5.20 - 5.23 the LSA and MLS-based retrieval relevancy and inverted index size were convergent).

The recall value of the MLS-based retrieval was the least in all the IR experiments (Exp1-Exp5): as shown in Figures 5.21, 5.23, 5.25 the recall values did not exceed r6 (it was less than 70%) whereas the LSA-based retrieval reached r7 in Exp 3 and JAC-based retrieval reached r8 in the experiments 2 and 3. The synonyms expansion in Exp 4 and 5 did not succeed in moving the recall to the upper recall level (only 1% improvement in Exp 4 and 5).

The small improvement in the relevancy obtained from the NBDV expansion: As appeared in Figures 5.25 and 5.26, 1% enhancement is achieved in both recall and MAP. But, the MLS extraction does not hurt the relevancy results, which impose a large remedy. In Exp 1, the difference in the MAP between the MLS-based retrieval and the MC-based retrieval is 4%, and the NBDV synonyms expansion reduced it to 3%; the difference in the recall was 13% reduced to 12%. The most important issue here is that the expansion does not support the recall at the expense of precision, which would occur if the expanded synonyms do not have any semantic relation to the terms that appeared in the user query.

5.5 Evaluation Chapter Summary

The evaluation chapter focussed on the evaluation of the results obtained in the experiments that have been applied to measure the performance of the MLS and NBDV models. Efficiency and relevancy measures were considered in the evaluation chapter, and comparisons with existing models were performed. Also, the evaluation chapter discussed the evaluation outcomes and stated the advantages and disadvantages of the developed models and also discussed the factors that affected the evaluation outcomes.

The next chapter will draw the final conclusions of the thesis after performing the evaluation process. All the evaluation outcomes will be summarized in final conclusions about the effectiveness of the developed text extraction models. The conclusion chapter will give the final statements. It will state if the MLS and NBDV models are efficient and effective based on the facts drown from the evaluation analysis and assessment.

CHAPTER 6 CONCLUSION and FUTURE WORK

An enhanced methodology to solve the problem of text overload in information retrieval is proposed, designed, implemented, and experimented. The methodology used semantic text analysis methods to extract the main ideas found in the text, which leads to short and informative summaries, and these summaries are used to build the inverted index in the IR system. The developed semantic analysis method uses a multi-layer approach of statistical analysis that investigates the verbatim overlaps in the lowest layer, the statistical calculations based on the VSM model in the middle layer, and the semantic meanings based on the latent semantic analysis in the upper layer. Based on the inverted index that was generated from the summaries of the documents, an IR system was designed and implemented. The IR system hired a traditional VSM model for matching the user query terms with the summaries terms. Also, the IR system was boosted by a synonyms extraction method called NBDV that efficiently and automatically extracts the synonyms of the query terms and appends them to the query before the matching process is initiated.

Both intrinsic and extrinsic evaluation approaches have been used to evaluate the extraction methods developed in this research. The intrinsic evaluation aimed to experience the MLS text extraction and the NBDV synonyms extraction as standalone methods in their fields. And, the aim of the extrinsic approach was to measure the influence of the MLS and NBDV methods on the relevancy and efficiency of the IR system. Three Arabic language datasets and one English dataset have been used in a series of experiments to evaluate the developed methods.

This chapter draws the final conclusions of the evaluations and analysis processes that were performed in my thesis. The conclusions in section 6.1 represent that achievements appeared during the intrinsic evaluation, whereas section 6.2 represents achievements appeared during extrinsic evaluation. In section 6.3 and 6.4, I revisited the objectives and the contributions of my thesis and showed with evidence how I achieved them. Section 6.5 describes the future plans of my research.

6.1 The Achievements Appeared during Intrinsic Evaluation

6.1.1 MLS Achievements

The MLS text extraction method presented an accurate text extraction method based on the use of an efficient semantic analysis framework. The method uses the centrality feature, and the centrality is computed using a multilayer statistical approach. The multilayer similarity computations designed to minimize the use of the LSA. To test our method of extraction, four entirely separated extraction systems have been built: (1) JacExtractor that is based on the Jaccard coefficient to measure the overlapped terms between two sentences, (2) VSMExtractor that is based on a traditional tf.idf scheme and VSM, (3) LSAExtractor that is based on the classical use of the LSA, and (4) MLSExtractor that is based on the semantic analysis framework proposed in this work[18].

Besides the ROUGE evaluation, we proposed a new evaluation technique based on the containment of the automatically extracted sentences in the manual extracts relative to the automatic extract size. The achievements of the MLS text extraction framework can be summarized in the following points:

The Multi-layer text extraction framework is effective: The analysis of the results showed that the proposed text extraction method was significant and succeeded in extracting a considerable ratio of the salient parts in the text. Depending on the containment evaluation, the four extraction methods succeeded in containing a high ratio of the sentences that appeared in the manual extracts. The percent of LOWC containment did not exceed 34% in all cases. Also, among the four extractors implemented in this research, the LSAExtractor and the MLSExtractor obtained significant results regarding the extraction quality and condensation rate, and this reflects the importance of investigating the semantic meaning of the text. JacExtractor and VSMExtractor obtained high results comparing with LSAExtractor results in terms of recall and Containment, but they failed to delete large portions of unnecessary text, and this appears clearly by scanning their CR values. On average, JacExtractor removed only 22 % of the original text and VSMExtractor removed 35%.

The Multi-layer text extraction framework is efficient: Our research showed that the MLS Extraction method remedies the time complexity problem related to the LSA extraction by (1) reducing the number of runs of the LSA similarity procedure and (2) reducing the original matrix dimensions. MLS extraction method decreased the number of executions of the LSA

[18] Parts of this section and its subsections are mentioned in the second paper in the "**Publications Arising from This Thesis**" section

program by 52% and the original matrix size by 65% and produced roughly the same ROUGE and Containment results obtained in the classical LSA extraction.

The Multi-layer text extraction framework is a strong competitor with stable behaviour: The other important conclusions that appeared after comparing our MLS extraction system with two existing extraction systems –UTF 8 SUPPORT tool and API tool- are the stability and accuracy. The MLSExtractor extraction system obtained higher ROUGE results than the UTF-8 SUPPORT and API extractors, and the generated extracts by the MLSExtractor had recall and precision values that are very close to their mean. The dispersion ratio was 8% for the recall values and 14% for the precision values.

The drawback that has been raised during the evaluation of the MLS method is the difficulty in identifying the boundaries of each layer because this depends on the contents of the text. We found that the text that is not semantically rich (no diversity in the vocabularies) gained fewer recall values. For example, the dataset of Exp 3 is a collection of posts of young people bloggers, and it contains the vocabularies that are used in everyday talk. In Exp3, the recall value decreased by 9% from the recall values recorded in Exp 2, which hired a semantically richer dataset.

6.1.1 NBDV Achievements

In this research, the NBDV synonyms extraction model is proposed, designed, and implemented. The method uses an unsupervised learning strategy to extract nouns synonyms. The NBDV substituted the traditional tf.idf weighting scheme with an efficient weighting scheme that weights the terms based on their semantic relation with the noun being processed. The targeted contribution of this research is to improve the efficiency, and at the same time, obtaining a significant precision. This contribution is achieved improving by the following achievements,

The NBDV is efficiency: the average number of terms needed to be processed for each run was 186 (instead of the processing of the whole terms found in the corpus). This average number is supported by a time complexity analysis that showed that the processing of each run of the NBDV method is accomplished in linear time.

The NBDV gives significant precision: the average precision that was evaluated based on well-known online dictionaries for the Arabic language was significant (51%), and this precision was proved by human experts who showed that 57.5 of the answer set contents are correct.

The discovered drawback of the NBDV was the low recall obtained for the ancient Arabic Language nouns. The ancient nouns are important because they found in AL Quran Al Kareem, which is the source of the Islamic religion. The system developed based on the NBDV method succeeded in returning only 36% of the synonyms found in the Almaany dictionaries, and this reflects the gap between the shallow vocabulary set used in the Arabic language media and the rich vocabulary set found in the Arabic language literature.

6.2 Extrinsic Evaluation Achievements

The main findings that are extracted after the employment of the MLS and NBDV models in the AIR process are summarized in the following points:

The MLS model has a positive influence on the efficiency of the AIR system without noticeable loss in the precision results. The size of the MSL inverted index is 58% smaller than the size of the original documents inverted index, which implies less time to match the index and the query terms and less space to store the index in the main memory and in the secondary storage devices. The precision relevancy measure in the five experiments that test the employment of the MLS in the IR system shows the convergent results between the MLS-based retrieval and the MC-based retrieval. The MAP of the MLS-based retrieval obtained 93% of the MAP obtained in the MC-based retrieval, and the recall-precision curves in the five experiments showed that the two curves that represent the MLS-based and MC-based retrievals were very close and the noticed difference appeared at high recall values (r6).

The NBDV model has a slightly positive impact on the relevancy of the AIR system. The influence of the NBDV synonyms expansion had a slightly positive impact (only 1% improvement in both recall and precision), but no negative impact has been recorded in all relevancy measures that are mentioned in Exp4 and Exp5. Note that the concluded time complexity of the NBDV method in the intrinsic evaluation was ***O(n)***, which does not hurt the time penalty of the whole retrieval process. Thus, the relevancy improvements came at no extra time requirements.

The negative impact of the MLS extraction on the IR system was the slight drop in the recall values. The relevancy evaluation of the MLS-based retrieval recorded 13% less recall than the MC-based retrieval (78% of the relevant documents retrieved by the MC-based retrieval, and 65% of the relevant documents retrieved by the MLS-based retrieval (Figure 5.21)). However, this ratio is considered sufficient if the time and space constraints are highly demanded.

6.3 The Research Objectives Revisit

The ten Objectives specified at the beginning of this research have been met. The following table summarizes the achievements of processing those objectives.

The objective	Achievements
Performing a precise survey that reviews the important publications in Arabic IR and provides a starting point for new researches in this field.	The survey showed that the researchers achieved significant enhancements in building accurate stemmers, with accuracy reaches 97%, and in measuring the impact of different indexing strategies. Query expansion and Text Translation showed a positive relevancy effect. However, other tasks such as NER and ATS still need more research to realize their impact on Arabic IR.
Setting a framework on how to employ the statistical semantic analysis based on the efficient use of latent semantic analysis in the text extraction.	The developed framework used multilayers of statistical analysis (MLS framework), and the LSA appeared at the upper layer. The applying of this framework showed that the LSA is necessary to process only 35% of the text, and the traditional statistical text analysis approaches which require less processing time can process the remaining text (Table 5.2).
Building an effective text summarizer using the efficient framework of semantic analysis.	The achievement of this objective was accomplished by the development of the MLS text Extraction method
Proving that the use of the traditional statistical bag of word models (such as theVSM and Jaccard coefficient) is not suitable for performing reasonable text summarization, especially to reduce the inverted index in an IR system.	The bag of Word models are used in (Perea-Ortega J. M.-L., 2013), (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001). The intrinsic evaluation in our research showed that these models failed to reduce the text size, and this appears clearly in the large values of the CR (68% and 79% respectively, Figure 5.5)
Improving the retrieval time through the reduction of the index size, which will be constructed from the summaries instead of the original documents.	The MLS-based retrieval constructed an inverted index that is 58% smaller than the main corpus inverted index. Figure 5.21, 5.23.
Analyzing the relevance of Information Retrieval systems with and without Automatic Text Summarization using IR evaluation measures.	All the relevancy results are collected in the Figures 5.20-5.27, and the final conclusion was mentioned in the Extrinsic Evaluation Findings section of this chapter.

Developing an efficient synonyms extraction model and employ this model in a synonyms extraction system that extracts synonyms for the user query terms.	The developed method is the NBDV synonyms extraction method, and the method processes the single synonyms extraction run in ***O(n)***.
enhancing the user query with the synonyms generated automatically and test their relevancy on the IR system that uses the summaries as a source of the index.	The outputted synonyms of the NBDV method are used to expand the user query, the recall and precision improved by 1%.
Estimating the effectiveness of our summarizers using extrinsic methods by evaluating their influence on Arabic information retrieval performance.	The summaries generated from the MLS extractor of Arabic language datasets were used to build the inverted index in information retrieval systems. The results showed a condensed version of the inverted index with comparable relevancy results with the original inverted index (All the figures appeared in Exp1,2,4,5)
Comparing the results of employing Arabic text summarization in information retrieval with previous results that have been obtained on other languages such as English.	Three previous publications addressed the use of summaries as a source of the index (Perea-Ortega J. M.-L., 2013), (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001). The primary difference between their work and our work was the method used to extract the summaries and the NBDV query enhancement. The relevancy assessment of our work that appeared in Figures 5.26, 5.27 was higher than their relevancy results that appeared in Table 2.4. And, the most important thing is the obtained relevancy results in our research is obtained in 42% CR, whereas, the relevancy results in the previous publication achieved at high values of CR.

6.4 Research Contributions with Evidence

The contributions that are mentioned in the introduction chapter have the following evidence from the obtained results[19]:

Contribution: Efficient and informative inverted index using a semantic-based text summarizer.

Evidence : MLS-based Retrieval

[19] Parts of this section and its subsections are mentioned in the second paper in the "**Publications Arising from This Thesis**" section

Comparing with the three research that used the summarization techniques to reduce the inverted index size (Brandow, Karl, & Lisa, 1995), (Sakai & Sparck-Jones, 2001), (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013), we obtained the following enhancements:

Significant Recall: As discussed in the introduction chapter, the previous work in this field obtained high precision and hurt the recall. For example, in (Brandow, Karl, & Lisa, 1995), the recall value declined by 41% (from 100% to 59%) and in (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013) the significant recall results obtained at 60%, 80%, and 90% condensation rate which implies that the reduction in the inverted index does not exceed 40%. In our work, we obtained higher recall that reached 66% in Experiment 4. and the difference in the recall between the MC-based retrieval and the MLS-based retrieval does not exceed 13% with a 58% reduction in the inverted index size(see Figures 5.21, 5.26).

Regarding Sakai and Sparck in (Sakai & Sparck-Jones, 2001), the authors experienced the summaries as a source of indexing for the precision-oriented search, so we cannot find recall measurements in there work.

Condensed and flexible size inverted index: In this research, the size of the inverted index is constrained by the salient information found in the documents, and we did not specify the condensation rate in advance. In spite of that, we reduced the inverted index size to 42%. The other publications in the field used a fixed condensation rate either as a fixed number of words (Brandow, Karl, & Lisa, 1995) or as a fixed ratio (Sakai & Sparck-Jones, 2001), (Perea-Ortega, Lloret, Ureña-López, & Palomar, 2013).

Contribution: Efficient framework for the semantic text analysis

Evidence: MLS model of text extraction

To remedy the time penalty of running the LSA in text extraction, we build an efficient framework that uses the LSA for certain parts of the text. Those parts represent that text segments that do not have a verbatim or statistical resemblance. The results of the developed framework (appeared in section 5.2) were comparable with the existing extraction system that used the classical LSA semantic analyzer.

Combining the information from figures 5.5 and 5.6 and Table 5.2, we found that the MLS method used the LSA for only 35% of the original text and obtained significant accuracy results as shown below:

	LSA - Text Extraction	MLS-Text Extraction	
P	41%	40%	
R	60%	47%	Figures 5.5, 5.6
CR	54%	42%	
The ratio of text processed by the LSA	100%	35%	

Contribution: Semantic representation of the text segments.

Evidence: The Deletion process in the MLS model (section 3.3.2)

As described in chapter 1, the centrality feature measures the importance of a certain segment of the text to the other segments. The MLS model computes the centrality of the text segment based on the semantic meanings of the vocabularies found in that segment. The MLS model uses an efficient semantic model to determine the centrality value. The centrality is determined by combining the vocabulary overlaps with the VSM and LSA models in a multilayer similarity scheme. The centrality feature is the only condition that controls the deletion of similar sentences in our deletion process (section 3.2.3), and the results of using the centrality feature were promising (at CR = 42%, AR = 48%, AP = 40% Figure 5.6).

Contribution: Robust evaluation strategy

Evidence: Containment Evaluation

In this research, the ROUGE intrinsic assessment of the Jaccard based extraction and the VSM based extraction gave higher results than the LSA analysis (Jaccard R=79%, VSM R = 63%, and LSA R = 60% Figure 5.6) which seems illogical and incompatible with the vast majority of research in this field. ROUGE evaluation cannot judge accurately the percent of complete sentences that are shared between the automatic and reference extracts, and it does not consider the size of the extract. Using the Containment evaluation, we found that 97% of Jaccard extracts sentences that are found in the manual extracts but at CR=79%, and we found that our method of extraction achieved 65% containment (65% of the manual extracts sentences appeared in the automatic extracts) at CR=42% (Figures 5.1- 5.4) . Note that ROUGE gave us misleading results and our evaluation gave a more fair judgment.

Contribution: Efficient and accurate semantic-based synonym extraction.

Evidence: NBDV model of synonyms extraction

Comparing with the bag of words models that are discussed in section 2.4, the NBDV model is more efficient because it generates the synonyms set in linear time. The time complexity analysis of the NBDV model that was held in section 5.3.3 showed that the time complexity of running the NBDV is O(n). Regarding the synonyms set accuracy, the automatic and manual evaluation showed significant precision and recall:

Automatic evaluation: Table 5.4

- R 47%
- P 51%

Manual Evaluation: Table 5.4

- P 57.5

In this research, the purpose of developing the NBDV model was to expand the query terms with semantically related words during the retrieval process. The improvements of 1% on both recall and precision in the MLS-based retrieval came without an additional time penalty. To proof that, assume that the number of terms in the whole corpus is n, and the number of terms in the inverted index is t with p postings associated with each t, then the time analysis of retrieval process will be:

- Scanning the inverted index requires q * t where q is the number of terms in the query.
- Retrieving the posting list elements requires p.

Therefore, Computing the similarity between the query and the retrieval posting lists requires in the worst case q * t * p. Note that q is a very small integer number, and the complexity can be written as O(t * p).

Knowing that the time complexity of the NPV method is O(n), the total time complexity of the retrieval process with NBDV synonyms expansion is:

O(n) + O(t * p)

Even if n is greater than t (because n represents the whole number of terms in the corpus with repetition), the value of t is multiplied by p, and p is not a small number, it may reach the number of documents found the whole corpus. This implies that the final time complexity is O(t * p).

6.5 Future Work

In the MLS extraction model, we used the Jaccard coefficient and VSM statistical techniques in the first and second layers. We cannot pretend that these two techniques are optimal, and in the future, we need to find the best statistical models that should be used in the first and second layers.

Also, the evaluation of the influence of the MLS text extraction on the relevancy of the other natural language processing applications, such as the text classification and Question Answering, should be measured. The purpose is to test if the replacement of the original text by the summaries generated from the MLS model will not hurt the accuracy of such applications.

Regarding the NBDV method, we plan to update the OWS weighting scheme to be able to process the word orders that end with verbs (OSV, SOV). If such an update is performed, then the NBDV becomes completely language independent. Also, measuring the effect of the OWS schemes in other text mining applications such as the named entity recognition, pattern recognition, and information retrieval is important to generalize this scheme. If the obtained results resemble the results achieved in the synonyms extraction field, then the OWS can replace the traditional tf.idf weighting scheme in the text mining applications, which will improve the efficiency of such applications.

References

Ababneh, M., Kanaan, G., Al-Shalabi, R., & Al-Nobani, A. (2012). Building an effective rule-based light stemmer for Arabic language to improve search effectiveness. *International Arab Journal of Information Technology (IAJIT), 9*(4).

Abbache, A., Barigou, F., Belkredim, F. Z., & Belalem, G. (2016). The Use of Arabic WordNet in Arabic Information Retrieval. *Business Intelligence: Concepts, Methodologies, Tools, and Applications. IGI Global*, 773-783.

Abdel Fattah, M., & Ren, F. (2008). Probabilistic Neural Network Based TextSummarization. *International Conference on Natural Language Processing and Knowledge Engineering.* Beijing.

Abdelali, A., Cowie, J., & Hamdy, S. S. (2007). Improving Query Precision Using Semantic Expansion. *Information Processing and Management, 43*(3), 705-716.

AbdelFattah, M., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM Based Models for Automatic Text Summarization. *Computer Speech & Language, 23*(1), 126 - 144.

Abderrahim, M. A., Mohammed, D. M.-A., & Mohammed, A. C. (2016). Semantic indexing of Arabic texts for information retrieval system. *International Journal of Speech Technology, 19*(2), 229-236.

Abderrahim, M., Mohammed, E. A., & Chikh, M. A. (2013). Using Arabic wordnet for semantic indexation in information retrieval system. *arXiv preprint arXiv, 1306*(2499).

AbdulJaleel, N., & Larkey, L. S. (2003). Statistical Transliteration for English-Arabic Cross Language Information Retrieval. *Proceedings of the twelfth international conference on Information and knowledge management* (pp. 139-146). ACM.

Abu-Salem, H., & Philip, K.-F. C. (2006). English-Arabic Cross-Language Information Retrieval Based on Parallel Documents. *International Journal of Computer Processing of Oriental Languages, 19*(1), 21-37.

Abu-Salem, H., & Philip, K.-F. C. (2006). English-Arabic cross-language information retrieval based on parallel documents. *International Journal of Computer Processing of Oriental Languages, 19*(1), 21-37.

Ageishi, R., & Miura, T. (2010). Automatic Extraction of Synonyms Based on Statistical Machine Translation. *2010 22nd IEEE International Conference on Tools with Artificial Intelligence* (pp. 313 - 317). Arras, France: IEEE.

Alhanini, Y., & Aziz, M. A. (2011). the Enhancement of Arabic Stemming by Using Light Stemming and Dictionary-Based Stemming. *Journal of Software Engineering and Applications, 4*(9), 522-526.

Aljlayl, M., & Frieder, O. (2001). Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation. *In Proceedings of the Tenth International Conference on Information and Knowledge Management* (pp. 295-302). ACM.

Aljlayl, M., & Frieder, O. (2002). On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach. *In Proceedings of the eleventh international conference on Information and knowledge management* (pp. 340-347). ACM.

Aljlayl, M., & Ophir, F. (2002). On Arabic search: improving the retrieval effectiveness via a light stemming approach. *Proceedings of the eleventh international conference on Information and knowledge management* (pp. 340-347). ACM.

Al-Kabi, M. (2013). Towards Improving Khoja Rule-Based Arabic Stemmer. *Proceedings of the 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, (pp. 1-6). Amman.

Al-Kabi, M., Kazakzeh, S. A., Abu Ata, B. M., Al-Rababah, S. A., & Alsmadi, I. M. (2015). A Novel Root Based Arabic Stemmer. *Journal of King Saud University - Computer and Information Sciences, 27*(2), 94-103.

Al-Kharashi, I. A., & Martha, W. E. (1994). Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science, 45*(8), 548-560.

Al-Kharashi, I. A., & Martha, W. E. (1994). Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System. *Journal of the American Society for Information Science, 45*(8), 548-560.

AlMaayah, M., Sawalha, M., & Abushariah, M. (2016). Towards an automatic extraction of synonyms for Quranic Arabic. *International Journal of Speech Technology,, 19*(2).

AL-Omari, A., & AbuAta, B. (2014). Arabic Light Stemmer (ARS). *Journal of Engineering Science and Technology, 9*(6), 702-716.

Al-Radaideh, Q. A., & Bataineh, D. Q. (2018). A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. *Cognitive Computation, 10*(4), 651–669.

Alruily, M., Hammami, N., & Goudjil, M. (2013). Using Transitive Construction for Developing Arabic Text Summarization System. *Computer and Information Technology (WCCIT), 2013 World Congress on 2013*, 1-2.

Al-Shalabi, R., & Kanaan, G. S. (2007). Stemmer Algorithm for Arabic Words Based on Excessive Letter Locations. *Proceedings of the 4th International Conference on Innovations in Information Technology. UAE: IEEE*, (pp. 456-460).

Al-Shalabi, R., Kanaan, G., Jaam, J., Hasnah, A., & Hilat, E. (2004). Stop-word removal algorithm for Arabic language. *Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on* (p. start page: 545). IEEE.

Al-Shalabi, R., Kannan, G., Hilat, I., Ababneh, A., & Al-Zubi, A. (2005). Experiments with the Successor Variety Algorithm Using the Cutoff and Entropy Methods. *Information Technology Journal, 4*(1), 55-62.

Alshamrani, H. (2012). Diglossia in Arabic TV stations. *Journal of King Saud University - Languages and Translation, 24*(1), 75-69.

Aone, C., Okurowski, M. E., & Gorlinsky, J. (1998). Trainable, Scalable Summarization Using Robust NLP and Machine Learning. *The 17th international conference on Computational linguistics: Association for Computational Linguistics*, (pp. 62-66). Stroudsburg.

Arabs, W. (2019, 3 3). *Arabs*. Retrieved 4 1, 2019, from wikipedia: https://ar.wikipedia.org/wiki/Arabs

Atwan, J., Mohd, M., Rashaideh, H., & Kanaan, G. (2016). Semantically enhanced pseudo relevance feedback for arabic information retrieval. *Journal of Information Science, 42*(2), 246-260.

Azmia, A. M., & Al-Thanyyan, S. (2012). A Text Summarizer for Arabic. *Computer Speech and Language, 26*(4), 260-273.

Baalbaki, R. (1988). *Al-Mawrid Dictionary Arabic-English.*

Ba-Alwi, F. M., Gaphari, G. H., & Al-Duqaimi, F. N. (2015). Arabic Text Summarization Using Latent Semantic Analysis. *British Journal of Applied Science & Technology, 10*(2), 1-14.

Babar, S., & Patil, P. D. (2015). Improving Performance of Text Summarization . *Procedia Computer Science, 46*, 354-363.

Baeza, Y., & Ribeiro, N. (1999). *Modern Information Retrieval.* Addison-Wesley Longman.

Baeza-Yates, R., & Ribeiro, B. (2011). *Modern information retrieval.* New York, Harlow, England: Addison-Wesley: ACM Press.

Barak, L., Dagan, I., & Shnarch, E. (2009). Text Categorization from Category Name via Lexical Reference. *NAACL-Short '09 Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume*, (pp. 33-36). Boulder.

Basha, M. (1992). *Alkhafi.*

Baxendale, P. (1958). Machine-Made Index for Technical Literature-an Experiment. *IBM Journal of Research and Development, 2*(4), 354 – 361.

Bellaachia, A., & Ghita, A.-T. (2008). Proper nouns in English–Arabic cross language information retrieval. *Journal of the American Society for Information Science and Technology, 59*(12), 1925-1932.

Benabdallah, A., Abderrahim, M., & Abderrahim, M.-A. (2017). Extraction of terms and semantic relationships from Arabic texts for automatic construction of an ontology. *International Journal of Speech Technology, 20*(2), 289-296.

Bessou, S., & Mohamed, T. (2014). n Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval. *Neural Network World, 24*(2), 117.

Bessou, S., & Touahria, M. (2014). an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval. *Neural Network World, 24*, 117.

Bessou, S., & Touahria, M. (2014). an Accuracy-Enhanced Stemming Algorithm for Arabic Information Retrieval. *Neural Network World, 24*(2), 117.

Binwahlan, M. S., Salim, N., & Suanmali, L. (2009). Intelligent Model for Automatic Text Summarization. *Information Technology Journal (Asian Network for Scientific Information), 8*(8), 1249 - 1255.

Bloehdorn, S., & Hotho, A. (2004). Boosting for text classification with semantic features. *In Proceedings of the MSW 2004 workshop at the 10th ACM SIGKDD conference* , (pp. 70–87).

Bloehdorn, S., Basili, R., Cammisa, M., & Moschitti, A. (2006). Semantic Kernels for Text Classification Based on Topological Measures of Feature Similarity. (pp. 808–812). Hong Kong: Data Mining, 2006. ICDM '06. Sixth International Conference on.

Boulaknadel, S., Daille, B., & Driss, A. (2008). Multi-word term Indexing For Arabic Document Retrieval. *2008 IEEE Symposium on Computers and Communications* (pp. 869-873). Marrakech: IEEE.

Bo-Yeong, K., Dae-Won, K., & Hae-Jung, K. (2005). Fuzzy Information Retrieval Indexed by Concept Identification. In *Text, Speech and Dialogue* (Vol. 3658, pp. 179-186). Springer Berlin Heidelberg.

Brandow, R., Karl, M., & Lisa, F. R. (1995). Automatic condensation of electronic publications by sentence selection. *Information Processing & Management, 31*(5), 675-685.

Chen, A., & Gey, F. (2002). Building an Arabic Stemmer for Information Retrieval. *University of California at Berkeley CA 94720-4600*.

Chen, H., & Lynch, K. (1992). Automatic construction of networks of concepts characterizing document databases. *IEEE TRANSACHONS ON SYSTEMS, MAN, AND CYBERNETIC, 22*(5), 885 - 902.

Chen, K.-Y. C., Liu, S.-H., Chen, B., Wang, H.-M., Jan, E.-E., Hsu, W.-L., & Chen, H.-H. (2015). Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23*(8), 1322-1334.

Chen, Q.-C., Wang, X.-L., Liu, B.-Q., & Wang, Y.-Y. (2002). Subtopic Segmentation of Chinese Document: an Adapted Dotplot Approach. *Proceedings of International Conference on Machine Learning and Cybernetics. IEEE, Pages. . :* , (pp. 1571 – 1576). Beijing.

Chen, Y.-L., & Chiu, Y.-T. (2011, May). An IPC-based vector space model for patent retrieval. *Information Processing & Management, 47*(3), 309-322. doi:10.1016/j.ipm.2010.06.001

Chowdhury, A., & McCabe, M. C. (1998). Improving information retrieval systems using part of speech tagging.

Christopher, D. M., Prabhakar, R., & Hinrich, S. (2009). *An Introduction to Information Retrieval.* Cambridge: Cambridge University Press, England, Online edition.

Crouch, C. J. (1990). An approach to the automatic construction of global thesauri. *Information Processing and Management, 26*(5), 629–640.

Dai, S., Diao, Q., & Zhou, C. (2005). Performance comparison of language models for information retrieval. *In IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 721-730). Boston, MA: Springer.

Darwish, K. (2002). Building a Shallow Morphological Analyzer in One Day. *Proceedings of the ACL-2002 Workshop on Computational Approaches to Semitic Languages.*

Darwish, K., & Oard, D. W. (2002). CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval. *In TREC. 2002. Gaithersburg.*

Deng, F., Stefan, S., & Sergej, Z. (2012). Efficient Jaccard-based diversity analysis of large document collections. *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1402-1411). ACM.

Dinh, D., & Tamine, L. (2015). Identification of concept domains and its application in biomedical information retrieval. *Information Systems and e-Business Management, 13*(4), 647–672.

Donga, T., Haidar b, A., Tomov b, S., & Dongarra, J. (2018). Fast SVD for Large-Scale Matrices. *Journal of Computational Science, 26*, 237–245.

Douzidia, F. S., & Lapalme, G. (2004). Lakhas, an Arabic Summarization System. *Proceedings of DUC'04*.

Duwairi, R., Al-Refai, M., & Khasawneh, N. (2007). Stemming versus light stemming as feature selection techniques for Arabic text categorization. *In Innovations in Information Technology, 2007. IIT'07. 4th International Conference* (pp. 446-450). IEEE.

Echeverry-Correa, J., Ferreiros-López, J., Coucheiro Limeres, A., Córdoba, R., & Juan Manuel, M. (2015). Topic identification techniques applied to dynamic language model. *Expert Systems with Applications, 42*(1), 101-112.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 264-285.

El-Beltagy, S. R., Rafea, A., & . (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems, 34*(1), 132-144.

El-Haj, M. O., & Hammo, B. H. (2008). Evaluation of Query-Based Arabic Text Summarization System. *2008 International Conference on Natural Language Processing and Knowledge Engineering, IEEE.* , (pp. 1-7). Beijing.

Elkateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. (2006). Building a WordNet for Arabic. *In Proceedings of the fifth international conference on language resources and evaluation (LREC 2006)*, (pp. 29-34).

Elshishtawy, T., & Al-sammak, A. (2009). Elshishtawy, T., Al-sammak, A.:Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques. *In: Proceedings of the Second International Conference on Arabic Language Resources and Tools CArabic IRo.*

El-Shishtawy, T., & El-Ghannam, F. (2012). Keyphrase Based Arabic Summarizer (KPAS). *Informatics and Systems (INFOS), 2012 8th International Conference: IEEE. NLP-7 - NLP-14.* Cairo: IEEE.

Evgeniy, G., & Shaul, M. (2007). Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization. *Journal of Machine Learning Research*, 2297-2345.

Fellbaum, C. (2005). WordNet and wordnets. *Encyclopedia of Language and Linguistics. Elsevier*, 665-670.

Fellbaum, C., & Vossen, P. (2007). Connecting the Universal to the Specific: Towards the Global Grid. *Lecture Notes in Computer Science, 4568*.

Fellbaum, C., & Vossen, P. (2012). Challenges for a multilingual wordnet. *Lang Resources & Evaluation, 46*(2), 313-326.

Ferreira, R., Cabral, L. d., Dueire, R. ,., Freitasa, S. F., Cavalcantia, G. D., Lima, R., . . . Favaro, L. (2013). Assessing Sentence Scoring Techniques for Extractive Text Summarization. *Expert System with Applications, 40*(14), 5755-5764.

Froud, H., Lachkar, A., & Ouatik, S. A. (2013). Arabic Text Summarization Based on Latent Semantic Analysis to Enhance Arabic Documents Clustering. *International Journal of Data Mining & Knowledge Management Process (IJDKP), 3*(1), 79-95.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques a survey. *Artificial Intelligence Review, 47*(1), 1-66.

Ghassan, K., Riyad, A.-S., & Sawalha, M. (2005). Improving Arabic Information Retrieval Systems Using Part of Speech Tagging. *Information Technology Journal, 4*(1), 32-37.

Ghawanmeh, S. A.-S. (2005). An Algorithm for Extracting the Root for the Arabic Language. *Proceedings of the 5th International Business Information Management Association Conference (IBIMA), CArabic IRo.* Egyp.

Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., & Rabab'ah, S. (2009). Enhanced algorithm for extracting the root of Arabic words. *n Computer Graphics, Imaging and Visualization, 2009. CGIV'09. Sixth International Conference* (pp. 388-391). IEEE.

Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., & Rabab'ah, S. (2009). Enhanced Algorithm for Extracting the Root of Arabic Words. *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*, (pp. 388-391). Tianjin.

Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., & Rabab'ah, S. (2009). Enhanced Algorithm for Extracting the Root of Arabic Words. *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference: IEEE*, (pp. 388 – 391). Tia.

Ghwanmeh, S., Kannan, G., Riyad, A.-S., & Ahmad, A. (2007). An Enhanced Text-Classification-Based Arabic Information Retrieval System. *2007 Innovations in Information Technologies (IIT)*, (pp. 461 - 465).

Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. *Linear Algebra*, 134-151.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery.* Springer US.

Guha, S., Rastogi, R., & Shim, K. (2000). ROCK: A robust clustering algorithm for categorical attributes. *Information systems, 25*(5), 345-366.

Guo, J., Xu, G., Cheng, X., & Li, H. (2009). Named Entity Recognition in Query. . *In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in Information Retrieval.* Boston, MA, USA.

Hafer, M. A., & Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information storage and retrieval, 10.11*(12), 371-385.

Halteren, H. v., & Teufel, S. (2003). Examining the Consensus between Human Summaries: Initial Experiments with Factoid Analysis. *, HLT-NAACL-DUC '03 Proceedings of the HLT-NAACL 03 on Text summarization workshop. Nijmegen: Association for Computational Linguistics*, (pp. 57-64).

Hammouda, F. K., & Almarimi, A. A. (2010). Heuristic lemmatization for Arabic texts indexation and classification.

Hanandeh, E. (2013). Building an Automatic Thesaurus to Enhance Information Retrieval. *IJCSI International Journal of Computer Science Issues, 10*(1).

Harrag, F., Aboubekeur, H.-C., & Eyas, E.-Q. (2008). Vector space model for Arabic information retrieval—application to “Hadith” indexing. *Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference.* IEEE.

Harris, Z. S. (1954). Distributional Structure. *WORD, 2*(3), 146-162.

Harwath, D., & Hazen, T. J. (2012). Topic Identification Based Extrinsic Evaluation of Summarization Techniques Applied to Conversational Speech. *, 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Pages.* , (pp. 5073-5076). Kyoto.

Hassel, M. (2004). *Evaluation of Automatic Text Summarization.* Sweden: Licentiate Thesis Stockholm. Retrieved from http://www.csc.kth.se/~xmartin/papers/licthesis_xmartin_notrims.pdf.

He, Z., Deng, S., & Xu, X. (2006). A Fast Greedy Algorithm for Outlier Mining. *Ng WK., Kitsuregawa M., Li J., Chang K. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2006. Lecture Notes in Computer Science, Volume 3918. Sprin, 3918*.

Henriksson, A., Moen, H., Skeppstedt, M., Daudaravicius, V., & Duneld, M. (2014). Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *JOURNAL OF BIOMEDICAL SEMANTICS, 5*(6).

Hmeidi, I., Alshalabi, R., Al-Taani, A., Najadat, H., & Al-Hazimah, S. (2010). A Novel Approach to the Extraction of Roots from Arabic Words Using Bigrams. *Journal of the American Society for Information Science, 61*(3), 583-591.

Hmeidi, I., Kanaan, G., & Martha, E. (1997). Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. . *Journal of the American Society for Information Science., 48*(10), 867–881.

Hull, D. A., & Gregory, G. (1996). Querying Across Languages: A Dictionary-Based Approach to Multilingual Information Retrieval. *In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 49-57). ACM.

Jianqiang, L., Yu, Z., & Bo, L. (2009). Fully Automatic Text Categorization by Exploiting WordNet. l*ecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 5839*, pp. 1-12. AIRS 2009: Information Retrieval Technology.

Jing, H., Barzilay, R., McKeown, K., & Elhadad, M. (1998). *Summarization Evaluation Methods: Experiments and Analysis.* In AAAI Symposium on Intelligent Summarization.

Khoja, S. (2012). *Khoja Stermmer*. Retrieved Jan 2016, from http://zeus.cs.pacificu.edu/shereen/research.htm

Kiyoumarsi, F. (2015). Evaluation of Automatic Text Summarizations based on Human Summaries. . *Procedia - Social and Behavioral Sciences, 192*, 83-91.

Kumar, N., De Beer, J., Vanthienen, J., & Moens, M. F. (2006). Evaluation of information retrieval and text mining tools on automatic named entity extraction. *In International Conference on Intelligence and Security Informatics*, (pp. 666-667). Springer, Berlin, Heidelberg.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A Trainable Document Summarizer. *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval*, (pp. 68 – 73). Seattle WA.

Kwok, K. (1989). A neural network for probabilistic information retrieval. *ACM SIGIR Forum, 23*(SI), 21 - 30.

Larkey, L. S., AbdulJaleel, N., & Connell., M. (2003). What's in a name?: Proper names in Arabic cross language information retrieval. *ACL Workshop on Comp. Approaches to Semitic Languages.*

Larkey, L. S., Allan, J., Connell, M. E., Bolivar, A., & Wade, C. (2003). UMass at TREC 2002: Cross Language and Novelty Tracks. *In: The Eleventh Text Retrieval Conference (TREC 2002). Gaithersburg: NIST.*

Larkey, L. S., Ballesteros, L., & Connell, M. E. (2007). Light Stemming for Arabic Information Retrieval. *Arabic computational morphology*, 221-243.

Lee, Y., Paining, K., Roukos, S., Emam, O., & Hassan, H. (2003). Language model based Arabic word segmentation. . *In: the Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, (pp. 399-406). Sapporo, Japan.

Leeuwenberga, A., Vela, M., Dehdar, J., & Genabith, J. v. (2016). A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. *The Prague Bulletin of Mathematical Linguistics, 105*(1), 111–142.

Li, L., Forascu, C., El-Haj, M., & Giannakopoulos, G. (2013). Multi-document multilingual summarization corpus preparation, part 1: Arabic, English, Greek, Chinese, Romania. *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, (pp. 1-12). Sofia, Bulgaria.

Lin, C. (2001). *SEE*. Retrieved from http://www.isi.edu/cyl/SEE.

Lin, C. Y. (2004). Rouge: A Package for Automatic Evaluation of Summaries. *In Proceedings of the Workshop on Text Summarization branches Out (WAS 2004).* Barcelona.

Lin, C.-Y. (1995). Topic Identification by Concept Gene. *Proceedings of the Thirty-third Conference of the Association of Computational Ling*, (pp. 308-310). Boston.

Lin, C.-Y. (1997). *Identify Topics by Concept Signatures.* Technical report, Marina Del Rey: Information Sciences Institute.

Lin, C.-Y. (1999). Training a Selection Function for Extraction. *Proceedings of the eighth international conference on information and knowledge management.*, (pp. 55-62). New York.

Lin, D., Zhao, S., Qin, L., & Zhou, M. (2003). Identifying synonyms among distributionally y similar words. *In IJCAI* , 1492–1493.

Lin, J., & Chris, D. (2010). Inverted Indexing for Text Retrieval. In J. Lin, & D. Chris, *Data-intensive text processing with MapReduce* (Vol. 3, pp. 65-84). Morgan & Claypool Publishers.

Live Science. (2004). *54094-how-big-is-the-internet*. (Live Science) Retrieved 12 4, 2018, from Live Science: www.livescience.com

Lobanova, A., Spenader, J., Cruys, T. v., Kleij, T. v., & Sang, E. T. (2009). Automatic Relation Extraction Can Synonym Extraction Benefit from Antonym Knowledge? *2009 Proceedings of WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies.* , (pp. 17-20).

Lonneke, v. d., & Jorg, T. (2006). Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, (pp. 866–873). Sydney.

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development (IBM), 1*(4), 309 – 317.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 159-165.

Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Expert Systems with Applications, 38*(10), 12708-12716. doi:10.1016/j.eswa.2011.04.058

Luo, S., Yinglin, W., Xue, F., & Zhenda, H. (2018). A Study of Multi-label Event Types Recognition on Chinese Financial Texts. *4th EuroSymposium on Systems Analysis and Design* (pp. 146-158). Cham: Springer.

Mahmoud, R., Sanan, M., & Zreik, K. (2011). Improving Arabic Information Retrieval System using n-gram method. *WSEAS Transactions on Computers , 10*(4), 125-133.

Mallat, S., Anis, Z., Emna, H., & Mounir, Z. (2013). Method of lexical enrichment in information retrieval system in Arabic. *International Journal of Information Retrieval Research (IJIRR) 3, 3*(4), 35-51.

Mani, I. (2001). *Automatic Summarization.*

Mani, I., Bloedorn, E., & Gates, B. (1998). *Using Cohesion and Coherence Models for Text Summarization.* Reston: AAAI Technical Report.

Mansour, N., Haraty, R. A., Daher, W., & Houri, M. (2008). An auto-indexing method for Arabic text. . *Information Processing & Management, 44*(4), 1538-1545.

Marcu, D. (1998). Improving Summarization through Rhetorical Parsing Tuning. *Workshop on Very Large Corpora.ACL Anthology Network. Pages*, (pp. 206-215).

Mashechkin, I. V., Petrovskiy, M. I., Popov, D. S., & Tsarev, D. V. (2011). Automatic Text Summarization Using Latent Semantic Analysis. *Programming and Computer Software, 37*(6), 299–305.

Maurice, d. K. (2018). *worldwidewebsize*. Retrieved 12 2018, 5, from https://www.worldwidewebsize.com/

Meena, Y. K., & Gopalani, D. (2015). Domain Independent Framework for Automatic Text Summarization. *Procedia Computer Science, 48*, 722–727.

Meena, Y. K., & Gopalani, D. (2015). Domain Independent Framework for Automatic Text Summarization. . *Procedia Computer Science, 48*, 722–727.

Mei, J.-P., & Chen, L. (2012). SumCR: A New Subtopic-Based Extractive Approach for Text Summarization. *Knowledge and Information Systems, 31*(3), 527 – 545.

Mihalcea, R., & Ceylan, H. (2007). Explorations in Automatic Book Summarization. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague: Association for Computation Linguistics*, (pp. 380-389). Prague.

Mikolov, T., Chen, K. C., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv, 1301*(3781).

Miller, G., Beckwith, R., Fel, C., Gross, D., & Miller, K. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography, 3*(4), 235–244.

Minkov, E., & Cohen, W. W. (2012). Graph based similarity measures for synonym extraction from parsed text. *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing Association for Computational Linguistics*, (pp. 20-24).

Minkov, E., & Cohen, W. W. (2012). Graph based similarity measures for synonym extraction from parsed text. *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing Association for Computational Linguistics*, (pp. 20-24). Jeju, Republic of Korea.

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. *Mining text data*, 43-76.

Nenkova, A., & McKeown, K. (2012). A Survey of Text Summarization Techniques. (C. C. Aggarwal, & E. ChengXiang Zhai, Eds.) *Mining Text Data*, 43-76.

Ngoc, P. V., & Tran, V. T. (2018). Latent Semantic Analysis using a Dennis Coefficient for ,English Sentiment Classification in a Parallel System. *International Journal of Computers Communications & Control, 13*(3), 408-428.

Nie, J.-Y. (2010). *Cross-language information retrieval.* (Vols. 8,8). Morgan & Claypool.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). A Robust Clustering Algorithm for Categorical Attributes. *international multiconference of engineers and computer scientists*, (pp. 380-384). Hong Kong.

Patil, M., Thankachan, S. V., Shah, R., Hon, W. K., Vitter, J. S., & Chandrasekaran, S. (2011). Inverted indexes for phrases and strings. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 555-564). ACM.

Perea-Ortega, J. M., Lloret, E., Ureña-López, L. A., & Palomar, M. (2013). Application of Text Summarization Techniques to the Geographical Information Retrieval Task. *Expert systems with applications, 40*(8), 2966-2974.

Perea-Ortega, J. M.-L. (2013). Application of text summarization techniques to the geographical information retrieval task. *Expert systems with applications, 40*(8), 2966-2974.

Pierre-Etienne, G., & Guy, L. (2011). Framework for Abstractive Summarization Using Text-To-Text Generation. . *MTTG '11 Proceedings of the Workshop on Monolingual Text-To-Text Generation.*, (pp. 64-73). Strouds.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery., B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge university press.

Ramanujam, N., & Kaliappan, M. (2016). An Automatic Multi-document Text Summarization Approach Based on Naïve Bayesian Classifier Using Timestamp Strategy. *The Scientific World Journal Volume*, 1-10.

Raynaud, C., & Fluhr, C. (1995). Textual database lexicon used as a filter to resolve semantic ambiguity application on multilingual information retrieval.

Rayner, K., Elizabeth, S. R., Michael, M. E., Mary, P. C., & Rebecca, T. (2016). So Much to Read, So Little Time How Do We Read, and Can Speed Reading Help? *Sychological Science in the Public Interest, 17*(1), 4-34.

Ryding, K. (1991). Proficiency despite diglossia: A new approach for Arabic. *Modern Language Journal, 75*(2), 212-218.

Sakai, T., & Sparck-Jones, K. (2001). Generic Summaries for Indexing in Information Retrieval. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 190-198). ACM.

Salton, G., & McGill, M. J. (1983). Index construction. In I. t. Retrieval, *Salton, G.; McGill, M. J.* Online Ed, McGraw-Hil, New York.

Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval. .* McGraw-Hill.

Salton, G., Wong, A., & Chungshu, Y. (1975, Nov). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613-620.

Sankarasubramaniam, Y., Ramanathan, K., & Ghosh, S. (2014). Text Summarization Using Wikipedia. *Information Processing & Management, 50*(3), 443–461.

Schler, J., Koppel, M., Argamon, S., & Pennebaker, J. (2006). Effects of Age and Gender on Blogging. *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.*

Schütze, H., Christopher, D. M., & Prabhakar, R. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press.

Scott , S., & Matwin, S. (1998). Text classification using WordNet hypernyms. (pp. 38-44). In Proceedings of the COLING/ACL Workshop on usage of WordNet in natural language processing systems.

Senellart, P. P., & Blondel, V. D. (2004). Automatic Discovery of Similar Words. In *Survey of Text Mining , Clustering, Classification, and Retrieval.* Springer.

Shaalan, K., Al-Sheikh, S., & Farhad, O. (2012). Query Expansion Based-on Similarity of Terms for Improving Arabic Information Retrieval. *International Conference on Intelligent Information Processing. .* Springer, Berlin, Heidelberg.

Shams, R., Hashem, M., Hossain, A., Akter, S. R., & Gope, M. (2010). Corpus-based Web Document Summarization using Statistical and Linguistic Approach. . *Computer and Communication Engineering (ICCCE), 2010 International Conference. IEEE.*, (pp. 1-6). Kuala Lumpur.

Singh, J. N., & Dwivedi, S. K. (2013). A comparative study on approaches of vector space model in information retrieval. *In International Conference of Reliability, Infocom Technologies and Optimization*, (pp. 37-40). Noida, India.

Slamet, C., Atmadja, A. R., Maylawati, D. S., Lestari, R. S., Darmalaksana, W., & Ramdhani, M. A. (2018). Automated text summarization for Indonesian article using vector space model. *Slamet, C., Atmadja, A. R., Maylawati, D. S., Lestari, R. S., Darmalaksana, W., & Ramdhani, M. A. (2018, JaIOP Conference Series: Materials Science and Engineering. 288*, p. 12037. IOP Publishing.

Song, S., Huang, H., & Ruan, T. (2018). Abstractive Text Summarization Using LSTM-CNN based deep learning. (S. US, Ed.) *Multimed Tools Appl (2018)*, 1-19.

Sparck, J. K., & Galliers, J. R. (1995). Evaluating Natural Language Processing Systems: an Analysis and Review.

Subhashini, R., & Kumar, J. S. (2010). Evaluating the Performance of Similarity Measures Used in Document Clustering and Information Retrieval. *2010 first international conference on integrated intelligent computing* (pp. 27-31). Bangalore, India: IEEE.

Svore, K. M., Vanderwende, L., & Burges, C. J. (2008). *Using Signals of Human Interest to Enhance Single-document Summarization.* Technical Report, Association for the Advancement of Artificial Intelligence.

Tayal, M. A., Raghuwanshi, M. M., & Malik, L. G. (2017). ATSSC: Development of an approach based on soft computing for text summarization. *Computer Speech and Language, 41*, 214–235.

Thompson, P., & Dozier, C. C. (1997). Name Searching and Information Retrieval. *In: Proceeding of 2nd conference of Empirical Methods in Natural Language Processing EMNLP 199*, (pp. 134-140). Rhode Island.

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using Wikipedia. (pp. 713-721). Las Vegas: Proceeding KDD '08 Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. doi:.1145/1401890.1401976

Wang, Q., Xu, J. ,., & Craswell, N. (2013). Regularized Latent Semantic Indexing: A New Approach to Large-Scale Topic Modeling. *ACM Trans. Inf. Syst, 31*(1), 1-44.

Wang, Y., & Ma, J. (2013). A Comprehensive Method for Text Summarization Based on Latent Semantic Analysis. In N. L. Computing. Berlin Heidelb: Springer-Verlag.

Wasson, M. (1998). Using Leading Text for News Summaries: Evaluation Results and Implications for Commercial Summarization Applications. p*roceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2. Association for Computational Linguistics.*

Webb, S. (2007). The effects of synonymy on second-language vocabulary learning. *Reading in a Foreign Language, 19*(2), 120-136.

Wedyan, M., Alhadidi, B., & Alrabea, A. (2012). The Effect of Using a Thesaurus in Arabic Information Retrieval System. *Int. J. Comput. Sci, 9*, 431-435.

William, B. ,., & Ricardo, B. (1992). *Information Retrieval: Algorithms and Data structures.* Amazon.

Yang, R., Bu, Z., & Xia, Z. (2012). Automatic Summarization for Chinese Text Using Affinity Propagation Clustering and Latent Semantic Analysis. *Web Information Systems and Mining, Lecture Notes in Computer Science* .

Yanmin, C., Bingquan, L., & Xiaolong, W. (2007). Automatic Text Summarization Based on Textual Cohesion. *Journal of Electronics, 24*(3), 338-346.

Yaseen, Q., & Hmeidi, I. (2014). Extracting the roots of Arabic words without removing affixes. *Journal of Information Science, 40*(3), 376-385.

Yates, R. B., & Neto, B. R. (1999). *Modern Information Retrieval.* Addison-Wesley Longman.

Yeh, J.-Y., Hao-RenKe, Yanga, W.-P., & Meng, I.-H. (2005). Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis. *Information Processing & Management, 41*(1), 75-95.

Yousefi, A. M., & Hamey, L. (2017). Text Summarization Using Unsupervised Deep Learning. *Expert Systems with Applications, 68*, 93–105.

Zhang, L., Li, J., & Wang, C. (2017). Automatic synonym extraction using Word2Vec and spectral clustering. *Control Conference (CCC), 2017 36th Chinese* (pp. 5629 - 5632). Dalian: IEEE.

Zhou, D., Truran, M., Brailsford, T., Wade, V., Ashman, H., & (2012). (2012). Translation techniques in cross-language information retrieval. *ACM Computing Surveys (CSUR), 45*(1).